Bolt Beranek and Newman Inc.

AFOSR-TR- 79-0674

LEVEL

AD A069879

Report No. 3739

Development of Human Performance Models for Man-Machine System Simulation

Interim Scientific Report for the Period October 1, 1976 to September 30, 1977

D.C. Miller, C.E. Feehrer, R. Muralidharan, R.W. Pew, and S. Baron

October 1978

Prepared for:
Air Force Office of Scientific Research

79 06 12 008

READ INSTRUCTIONS BEFORE COMPLETING FORM

REPORT DOCUMENTATION PAGE

1. REPORT NUMBER AFOSR-TR-79-0674	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Development of Human Performance Models for Man-Machine System Simulation		5. TYPE OF REPORT & PERIOD COVERED Interim ~~Scientific~~ Report 1 Oct 76 - 30 Sept 77
		6. PERFORMING ORG. REPORT NUMBER BBN ~~Report No.~~ 3739
7. AUTHOR(s) Duncan C. Miller, Carl E. Feehrer, Ramal Muralidharan, Richard W. Pew, Sheldon Baron		8. CONTRACT OR GRANT NUMBER(s) F44620-76-C-0029
9. PERFORMING ORGANIZATION NAME AND ADDRESS BOLT BERANEK AND NEWMAN INC. 50 Moulton Street Cambridge, MA 02138		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61102F 2313A4
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Office of Scientific Research Bolling AFB DC 20332 (NL)		12. REPORT DATE October 1978
		13. NUMBER OF PAGES 75
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

man-machine modelling
human performance
modelling & simulation
optimal control theory
human reliability prediction
SAINT modelling

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report contains discussions and program flow charts pertinent to bottom-up and top-down models developed by BBN to predict the performance of RPV controllers. Included are brief discussions of the control task itself and of problems and issues encountered during model development.

infrequent and monitoring and decision-making are the operator's main tasks.

The task modelled is a simplified version of a simulated RPV mission. It retains many of the cognitive aspects of the full simulation but differs in several details, particularly with respect to the operator/system interface. The analysis of this problem illustrates some of the major considerations in applying DEMON to complex, supervisory control problems. It shows that with fairly straightforward assumptions about the operator's task, DEMON will give reasonable predictions of performance. However, the model results are not compared with actual data so DEMON is presently unvalidated.

The development of DEMON was part of a three year research program for the Air Force Office of Scientific Research aimed at investigating human performance models. The report also provides a brief summary of the overall effort.

Report No. 3739 Bolt Beranek and Newman Inc.

DEVELOPMENT OF HUMAN PERFORMANCE MODELS FOR MAN-MACHINE SYSTEM SIMULATION

Prepared by

Duncan C. Miller
Carl E. Feehrer
Ramal Muralidharan
Richard W. Pew
Sheldon Baron

Interim Scientific Report for the
Period Oct. 1, 1976-Sept. 30, 1977

Prepared under Contract F44620-76-C-0029

for

Air Force Office of Scientific Research

October 1978

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

The Systems Research Branch of the Human Engineering Division of the Aerospace Medical Research Laboratory has undertaken a long-term program to develop and exploit simulation and modelling technology in the design and evaluation of large-scale systems. In support of this goal, BBN has initiated, with AFOSR support, a three-year program of research to review human performance models and modelling technology for application to command and control systems. Our goal is to develop the beginnings of a handbook-like document that would be useful to systems designers and analysts embarking on a modelling effort. Our research program assumes that computerized models of the system under consideration are to be constructed, and that these models will take into account the behavior of the human decision-makers interacting with the system. By exercising these models, systems engineers can predict the effects of changes in system parameters before proceeding to full-scale simulation efforts and the operational evaluation of prototype systems.

During the first year of effort, we reviewed a rather extensive literature in human performance modelling, including data bank formulations, network-based techniques, control-theoretic models, information processing models, and some miscellaneous models having an operations-research flavor. From this review we distilled a set of issues concerning human performance modelling that needed to be addressed, and we recommended research that would contribute to the resolution of those issues. This work is documented in BBN Report 3446, entitled "Critical Review and Analysis of Performance Models Applicable to Man-Machine Systems Evaluation" (1977). The Executive Summary of this report is attached as Appendix E.

This Interim Scientific Report documents our work during the second year of the project. One conclusion of the first year's effort was that there were substantial philosophical and practical differences between two approaches to modelling. The **top-down** approach begins with overall goals and criteria of good performance and develops the assumptions about human and system performance that are necessary and sufficient to characterize performance in relation to the significant parameters of interest to system designers. The bottom-up approach begins with a detailed analysis of the tasks and subtasks required of the human operator and postulates component models to represent each subtask. These subtask models are then integrated logically to produce the overall task structure and to predict system performance. We also observed from the literature that in no case have alternative approaches, such as these, ever been applied to the same problem so that their strengths and weaknesses could be compared directly. We were interested both in the process of model development from these two perspectives as well as the relative usefulness of the products that result.

Accordingly, we have undertaken to develop an example model of each type for application to the representation of system performance of the enroute-control task of the RPV manned simulation. We have completed the initial formulation of the bottom-up model and have delivered to AMRL the flow charts and specifications required to integrate our model into the SAINT simulation of the RPV control task developed by Wortman, et al(1976). This initial model is described in Section 3 of this report.

The top-down model is being derived from the BBN Optimal Control Model. Early in the second year it was decided, in consultation with the COTR, that the best strategy for development of the top-down model would be to prepare a preliminary testbed at BBN, to develop the model in this context, and to undertake debugging and check out before delivering to AMRL an implementation suitable for integration into the SAINT simulation context. Additional funds have been made available to complete this development early in the third year. Section 4 of this report presents the current status of this model.

A third goal of the second year of effort was to extend our examination of the issues that must be addressed by system designers who embark on a program of model development as an aid to the system design process. In Section 5 of this report, we present a discussion of the issues that we have identified on the basis of our detailed development efforts on these two models. This discussion is preliminary and will be augmented and elaborated at the conclusion of the third year of work.

2. BACKGROUND

2.1 The RPV Control Problem

The task context in which the models to be described were built was the ground-based control of multiple flights of remotely-piloted vehicles. AMRL has developed a five-station manned simulation of this RPV drone control facility, and Pritsker & Associates, Inc. has developed a SAINT computer simulation of the performance of both the operators and equipment components of this facility. It was intended that the models to be built by BBN would replace certain elements of the Pritsker model. The description of the RPV control problem given here is, therefore, abstracted from Wortman et al (1976). The reader is referred to this report for a more complete description.

In the manned simulation, an RPV mission consists of coordinated flights of up to eleven groups of three RPV's, each group having one strike vehicle (S), one electronics countermeasures vehicle (E), and one reconnaissance vehicle (L). The S and E vehicles fly over the target 15 seconds apart, while the L vehicle follows two minutes later to assess damage.

At launch, each RPV is assigned a flight path that is assumed to be optimal in terms of terrain and defense. The vehicle is automatically controlled with respect to this flight path; however, each vehicle is subject to flight-path errors resulting from navigation system errors, position-reporting errors, communications jamming by the enemy, or equipment malfunctions. Because of these errors and resultant drifts off course, the vehicles require external monitoring and control from the ground station to keep them as close to the desired path as possible. This supervision is provided by four human en route controllers, who are equipped with CRT displays for monitoring of flight path and vehicle status and with keyboards and light pens for introducing changes in RPV flight parameters.

Strike RPVs are handed off to a terminal controller, who is equipped with a television picture of the view from the nose of the RPV and with standard aircraft controls and displays in order to direct each vehicle to a specific designated target, release its payload, and hand it back to one of the en route controllers. To simulate equivalent operations for E and L vehicles, the en route controllers hand off these vehicles to a **pseudo-pilot**, using the same procedures. The operator designated as pseudo-pilot receives a vehicle by operating a toggle switch on his control panel. At a specified time, these vehicles are handed back of the en route controllers at a designated location on their pre-defined flight path. The models we are developing address only the en route and return phases of the mission.

For strike vehicles, the flight path includes three **waypoints**. The S waypoint identifies the position at which the vehicle is prepared for handoff. The H waypoint designates the desired point of actual handoff to the terminal controller. Finally, the B waypoint designates the point at which the vehicle is handed back from the terminal controller to one of the en route-return controllers. For E and L vehicles, only the H and B waypoints are identified.

At the beginning of a simulated mission, the en route controllers first examine the pre-scheduled times that each strike vehicle is to arrive at handoff; they then generate, with paper and pencil, a revised schedule that spaces handoffs to be separated by two minutes so that overlaps in terminal control requirements do not occur. They also adjust the speed of one or more strike vehicles to meet this revised schedule.

During the remainder of the mission, the en route controllers are responsible for monitoring the flight path of vehicles assigned to them, for issuing commands correcting flight path and velocity, and for dealing with any contingencies that may arise.

In order to conduct these activities, they are provided with a listed/tabled summary status for all RPVs and with capabilities for displaying the flight path and detailed status of each vehicle. The entire simulation operates on a five-second frame update rate, so that displays are updated once each five seconds and commands are only implemented in synchrony with this update period. The status summary, which is displayed continuously, presents the vehicle number, estimated time of arrival at the next waypoint, and a three-character code that describes command link status, waypoint designator, and flight mode. In addition, a number is incremented automatically for each five-second period during which a given vehicle deviates from the prescribed flight path by more than an adjustable threshold. In order to examine the actual flight path, detailed vehicle parameters, or commands issued but not yet carried out, the operator must point to the RPV number in question on the status menu and depress a key on the special-purpose keyboard.

To enter a **patch** (a change in RPV flight path), the operator indicates the desired change by designating one or more points on the revised flight path, depressing the reconnect function key, and then designating the desired reconnect point. If the change does not violate turn-radius constraints, and if the command link is operational, the command will be executed at the next five-second frame update. Otherwise, the command will be rejected by the system and the operator will be so informed.

To enter a change in vehicle speed, the operator must indicate that a velocity change is required on the function keyboard, designate the RPV with the light pen, type in the new velocity on the standard keyboard, and depress the EOB key.

Just prior to the S waypoint, an S RPV is prepared for handoff by a **pop-up** maneuver that includes changing its speed to 250 knots and changing its altitude to 3000 feet using a procedure similar to velocity change with the altitude-change function key. Pop-up for E and L vehicles occurs just prior to the H waypointand involves an altitude change to 3000 feet and a velocity change to 400 knots.

The en route controllers are instructed that their highest priority is the timely execution of pop-ups, their second priority should be maintaining the desired ETAs and separations between S, E, and L RPVs, and their third priority should be to minimize flight path deviations.

2.2 Overview of Original SAINT/RPV Simulation

2.2.1 Model Structure

The original SAINT/RPV model has two primary components: (1) a state variable component, which consists of the simulation of RPV flight position, navigation system errors, maneuverability constraints, fuel consumption, effects of disturbances on flight, and the impact of operator commands; and (2) a discrete task component, which simulates the sequence of control, decision, and other operator tasks reviewed in Section 2.1 that must be performed in carrying out the RPV mission.

With a few exceptions, all operator tasks defined in the SAINT/RPV simulation share the following characteristics:

(1) They can be performed by any one of the four operators on the control team.

(2) The times required for their performance are selected from specified distributions, most frequently normal, and are rounded off to the nearest five-second interval. All elapsed times employed are equal to or greater than zero seconds and less than 9,999 seconds.

(3) They are equal in priority.

The SAINT/RPV simulation model embodies a number of mechanisms that are required for coordination between the state-variable and task-oriented components of the model, for computation of task times, and for matching of simulated operator performance to that exhibited by real operators. One such mechanism is the **Operator Attribute** file, which provides a means for representing individual differences among operators with respect to decision thresholds and criteria. Following Wortman et al (1976), a short catalog of such factors is as follows:

"(1) The time before the RPV reaches its handoff coordinates that the operator prefers to initiate the pop-up maneuver;

(2) The times before the RPV reaches its handoff coordinates that the operator prefers to make a velocity change, the altitude change, and the handover to the terminal pilot or pseudo-pilot;

(3) The lateral deviation value for an RPV above which the operator will make a directional change for that RPV; and

(4) The difference between the actual ETA and the desired ETA of an RPV that the operator deems acceptable." (p.40)

Values for each of the twenty-two Operator Attributes defined in the program are input for each operator on the team prior to a run of the simulation. As each task is initiated during the run, the program determines which operator will be responsible for its execution, and then acquires the values of the attributes that characterize the identified operator's performance.

2.2.2 Operator Task Sequences

A simulated RPV mission begins with each operator monitoring the progress of RPVs assigned to him. He then determines whether or not one of the vehicles has reached the point at which he prefers to pop it up. If so, the pop-up procedure is executed and the operator then waits until it is time to hand the RPV off to another operator. After handoff, the operator waits until the RPV has been flown through the target area by the terminal pilot and has been handed back. He then pops the RPV down for the return leg of the mission.

If no pop-up or pop-down procedures are called for, as would be the case during early and late stages of a typical mission, the operator determines whether or not any of his RPVs are malfunctioning. Any malfunctions are corrected, if possible, and the operator turns to consideration of whether or not the velocity of one or more of his RPVs should be changed in order to minimize errors in arrival time at the handoff point.

When necessary adjustments in velocity have been completed, the operator decides whether or not the flight path of any of his RPV's requires amendment (or **patching**). If so, and if all constraints relative to the current position of the RPV are satisfied (e.g., it is not near a programmed turning point), the operator proceeds to input a change in the flight path.

Returning RPVs are checked to determine the adequacy of their fuel supplies, and velocities and altitudes are modified by the operator to conserve fuel when necessary. The operator then returns to the monitoring function and the process begins again.

The original SAINT/RPV simulation was designed to replicate the organization and performance of a particular team of controllers during a particular run of the RPV II series of experimental missions. To achieve this goal, several modifications to the general character of operations outlined in preceding paragraphs were introduced. The most significant of these relate to (1) specialization of operator responsibilities and (2) pre-programed hand-off failures and other missed operations during the mission. Some of these modifications are noted briefly below.

(1) Operator specialization. During the mission being replicated, the team organized itself as follows. During the initial enroute phase of the mission, the operators divided the RPVs equally, with each operator being responsible for one flight of three (an S, an E, and an L) and one or two others. Operator 1 took responsibility for handing off all S RPVs. Operator 2 handed off all E RPVs (except one), and operator 3 handed off all L RPVs (except one). Operator 4 handed off one E and one L, and was then responsible for carrying out fuel checks on all RPVs after they were on the return leg of the mission. This particular organization was reproduced in the SAINT/RPV simulation.

(2) Missed handoffs and popups. When a particular operation did not occur in the mission being replicated, appropriate entries were made in an array of missed operations that served as initial conditions for the simulation. This, of course, insured that the same failures would take place in the simulation as in the original mission.

(3) Idiosyncratic controller behavior. Certain controllers apparently neglected particular RPVs during the mission being replicated. This behavior was accounted for within

the SAINT/RPV simulation by suppressing consideration of particular RPVs entirely during certain time periods, and randomly neglecting them during other time periods.

(4) Variable team activity levels. During non-critical phases of the mission being replicated, the controller team apparently relaxed and issued only occasional commands. This behavior was reproduced in the simulation by sampling from an exponential time delay distribution as the "monitor mission status" task was executed.

2.3 Interfacing Revised Models with SAINT

In revising the components of a simulation model of the size and complexity of the present one, we must strive to minimize unnecessary changes. The models outlined in Sections 3 and 4 of this report reflect this philosophy. It is not enough, however, simply to refrain from directly affecting model components without a compelling reason. It is also essential to avoid, as far as possible, indirect effects on one component caused by modifications to another. As we note in Section 5 of this report, avoiding unintended interactions necessitates the investment of sufficient time to achieve a fairly thorough understanding of the original model, its structure, and its components.

2.3.1 Bottom-up Models

In evaluating the original SAINT/RPV model components, we concluded that the areas most in need of revision were those associated with system monitoring and decision-making, rather than those associated with carrying out decisions once they have been made. Fortunately, the original model segregates tasks fairly cleanly along these dimensions.

The overall structure of the revised bottom-up model is presented in Section 3.2. Briefly, the approach employed was to merge the tasks involved in the primary monitoring and decision loop into a single, centralized task, while leaving the tasks involved in executing the decisions essentially unchanged. The revised model should therefore be highly compatible with the original model. Some changes will be required in the table of possible successor tasks and in the Operator Attribute list. Otherwise, all changes will occur within the FORTRAN CODE of a few MODRFs and USERFs. Appendix C of this report contains a list of changes that will be necessary. In constructing this list, we have carried out a thorough cross-check of the implications of changes in each task on all other tasks, and we are reasonably sure that we have identified all major interactions among the model components.

2.3.2 Top-down Models

For the top-down model, an additional set of factors comes into play. Since this model is structured around BBN's Optimal Control Model of the human controller, it must have access to periodic samples of the system state variables and must have some knowledge of their statistics. The decision-making components of the original SAINT/RPV simulation, however, are asynchronous or "event-oriented."

The differences between a synchronous, sampled-data model and an asynchronous, event-oriented model can be profound. In the present case, however, it is fortunate that some basic components of the SAINT/RPV model are driven by events associated with the periodic 5-second frame updates, and hence are essentially synchronous themselves. One such component is the STATE subroutine, and it is this subroutine that updates the real and virtual flight plan positions of the RPVs.

We believe, therefore, that it will prove possible to implement the top-down model in such a way that it can simply replace the centralized monitoring and decision task that we have developed as part of our bottom-up approach. Under this strategy, the top-down model would (at least in its initial implementation) release the same command-generation and command-execution tasks as the bottom-up model. In later implementations, some of the latter tasks, particularly the velocity-change and patch-generation tasks, may be drawn inside the central Optimal Control Model. See Section 4 for a more detailed picture of the modular construction of the top-down model.

An additional advantage of structuring the top-down and bottom-up models with an interchangeable central decision-making module is that this approach will permit a direct comparison of the two approaches without the confounding effects of differences in other model components.

3. BOTTOM-UP APPROACH

3.1 Rationale for Revised Approach

A fundamental purpose of computer simulations of human performance is to permit the exploration of alternative system designs without the necessity of actual implementation. To achieve this purpose, the models employed must be valid over the range of configurations and system parameters of interest. They must also be predictive in the sense that they must be specifiable in advance of the collection of specific human performance data for the system being simulated. Finally, they must be formulated in such a way that changes in system parameters can be taken into account without affecting the underlying structural features of the models.

The original SAINT/RPV simulation falls short of these requirements on several counts. As noted in the previous section, it took advantage of human performance data collected within the very context being modelled, and it utilized "foreknowledge" of certain events that could never be available in a truly predictive model. As was also noted in the previous section, the decision structures and monitoring strategies employed were mechanical, and did not reflect the kinds of priority judgments that humans perform so well.

One specific aspect of the model that needs to be revised is the parameter search process that the model employs. In the original model, the operator cycles through all RPVs for which he is responsible, evaluating performance with respect to a single mission parameter. If he finds any discrepancies that require action, he interrupts his serial search to deal with the problem, and then resumes his search with the next RPV. This "RPV-by-parameter" search paradigm does not represent the kinds of strategies normally employed by human controllers. It fails to take into account the fact that experienced operators continually reorder their priorities as tradeoffs among critical dimensions appear in the course of task performance. For example, even though ETA deviations at handoff are considered more critical than accumulated flight path errors, a very large flight path deviation may be more critical at a given point in a mission than a small ETA error. Another example of a situation in which priorities must be reordered occurs when one or more RPVs are critically close to a handoff point, and it is questionable whether sufficient time remains for correction of a problem. Under these conditions, a controller might elect to forego an attempt at correcting the RPV near handoff, and might concentrate instead on rectifying deviations for the next most critical RPV.

In developing revised models, we have attempted to utilize components that are truly predictive in nature, broad in scope, and flexible enough to reflect the human controller's ability to dynamically reorder priorities. The remainder of Section 3 will describe our revisions.

3.2 Overview of Revised Approach

BBN's approach to modelling RPV controller performance differs from the original approach in three important respects: (1) instead of searching one parameter at a time, it utilizes a paradigm in which all the information available for a given RPV is extracted before the next RPV is considered; (2) it introduces a deferred action concept in which the simulated controller postpones the taking of corrective action with respect to an RPV of low priority if an RPV of higher priority requires correction, and then returns attention to the deferred item when time is available; and (3) it avoids the use of "regression models" with parameters that must be determined experimentally within a particular application, and utilizes models with greater generality for the prediction of controller performance.

The key element of the revised model is the monitoring loop shown in Figures 1 and 2. This loop replaces tasks 91, 8, 10, 13, 16, and 18 of the original model, and combines their functions into a single task, which we have denoted as "New Task 13". Table 1, which follows the flowcharts, summarizes the key variables that appear therein. For additional information regarding the SAINT/RPV variables, see Wortman et al (1976).

In formulating the revised model, we have assumed that all operators are identical in their behavioral characteristics. A particular decision will depend on the types and states of the RPVs being controlled by an operator, but it will not depend on which operator is involved. This approach will permit the model to be used for a limited exploration of the effects of different "specialization" strategies employed by various teams of operators. Note, however, that some of the most crucial effects of these specialization strategies -- reduced confusion among differing handoff procedures, inter-operator coordination and communication, etc. -- remain unmodeled. Until the impact of these factors has been assessed and appropriate additions made to the revised model, results obtained with this simulation must be interpreted with caution.

The revised model is designed to reflect the complex priority structure that the operators must employ. Some types of deviations are inherently more serious than others, but a large deviation on a low-priority dimension can be more critical than a small deviatic on a high-priority dimension. Moreover, the importance of a given deviation will often be a function of how much time is available in which to correct it. As a first approximation to this priority structure, the model includes two sets of action limits. The first set is termed **immediate-action** limits, and consists of those values of various state deviations that will cause the operator to institute an immediate correction. The second set is termed **deferred-action** limits, and represents the values that the operator will employ if he finds no deviations that exceed the immediate-action limits. Both sets of limits depend, in general, on RPV type, mission phase, and time remaining before the next waypoint. The revised model is structured as a two-pass process. During the first pass, the operator checks each RPV against the immediate-action limits in order of descending priority. If no deviations exceeding these limits are found for any RPV, he proceeds to a second pass, employing the deferred-action limits.

As in the original model, each operator has an "en route/return" list of RPVs for which he is responsible during most of the mission, and a "terminal area" list of RPVs which he prepares for handoff to the terminal area pilot and which he receives back when the RPV has cleared the terminal area. These two lists may be identical under some organizational schemes, while under others they might be completely different.

Upon entering new task 13, the operator first checks his terminal area responsibility list to determine whether there are any RPVs that are close to the points at which they must be popped up. If so, he checks to see whether the pop-up must be initiated immediately or whether there is time to carry out other checks within Task 13. If insufficient time remains, he proceeds to Task 27 to perform the pop-up; otherwise, he continues checking his terminal-area list.

If no pop-ups are imminent, he checks his terminal-area list again for RPVs that have been handed back by the terminal pilot and are ready for pop-down. Upon finding one, he proceeds to Task 43 to perform the pop-down; otherwise, he continues checking his terminal area list. During this phase of Task 13, he also checks for unacceptable lateral deviations for S RPVs that have been popped up at S, but that have not yet reached H. If such a deviation is found, he proceeds to Task 52 to correct it.

If no required activities are discovered in checking the terminal-area list, the operator proceeds to his en route/return list. Beginning with the first RPV on his list that is still enroute, he checks each RPV in turn for required reroutes, reprograms, and malfunctions (all of which are initiated by other tasks of the original model), and then for unacceptable ETA errors and LATDEV errors. He first checks for serious errors, using the "immediate-action" error limits. If no such errors

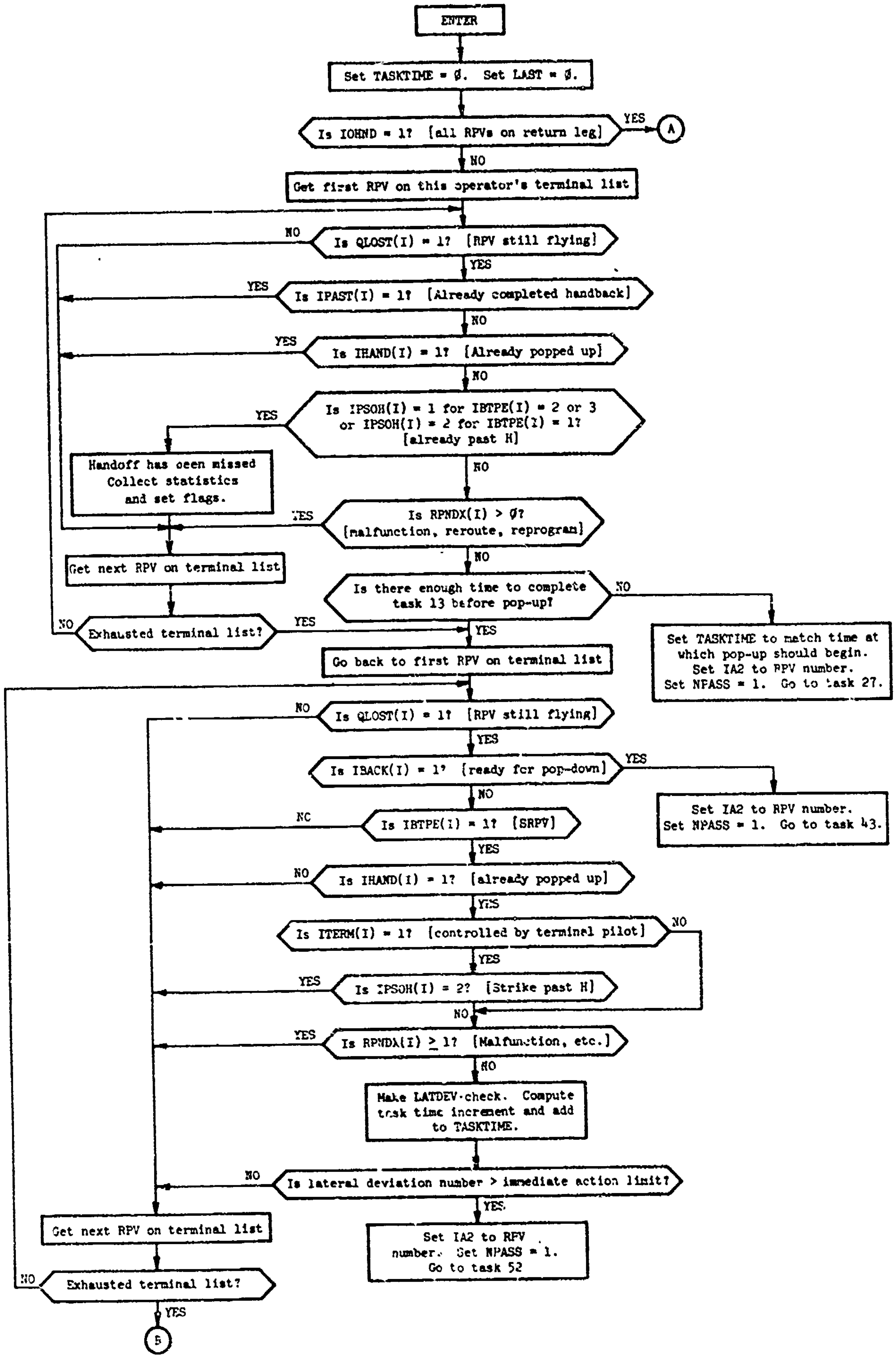

Figure 1. Decision processes used for terminal area list.

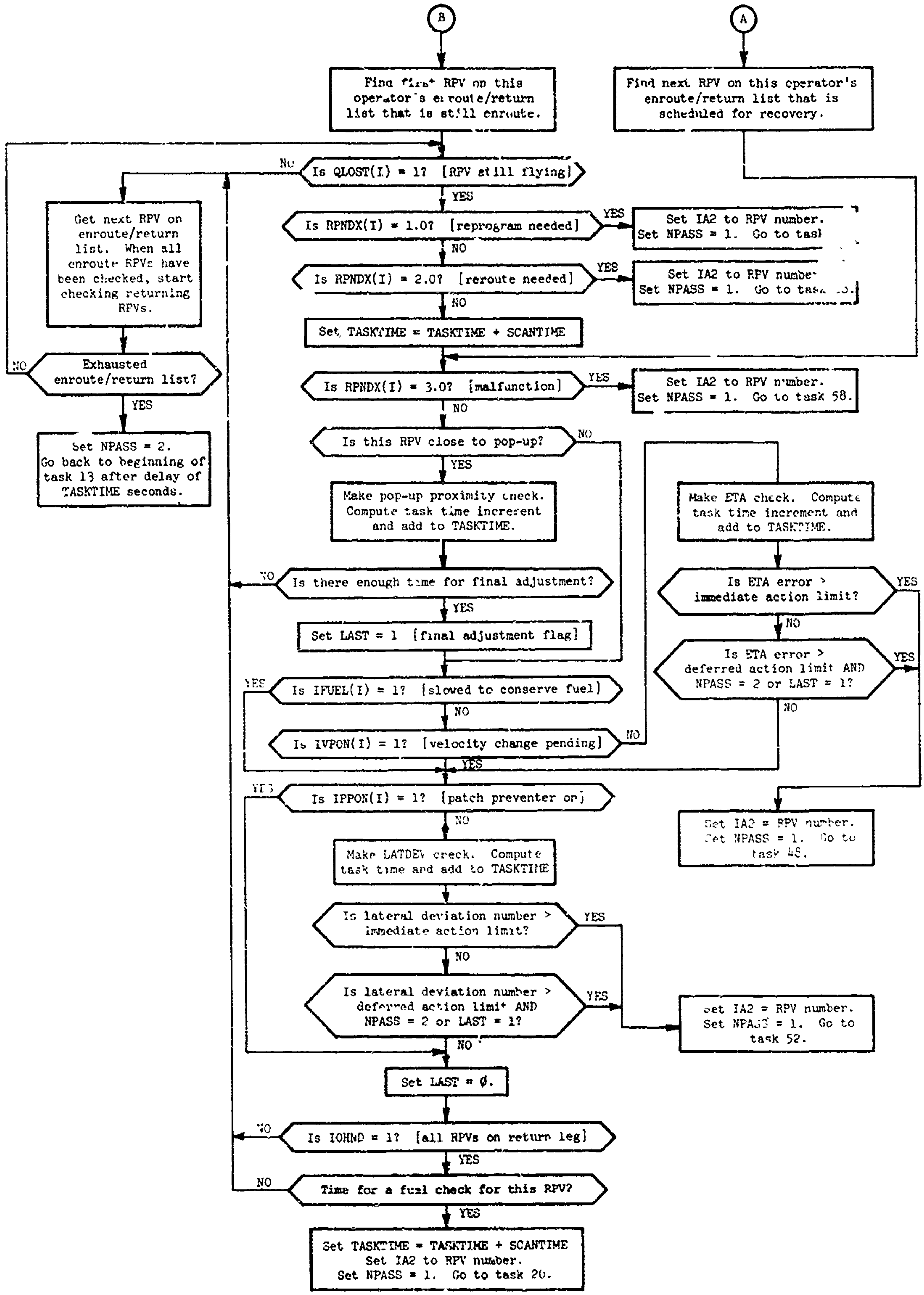

Figure 2. Decision processes used for en route/return list.

Table 1. SUMMARY OF KEY VARIABLES USED IN NEW TASK 13

IA2	Information attribute 2. An element of a vector used within SAINT to communicate information regarding the RPV to which a command refers.
IBACK(I)	Unchanged. Handback index for RPV I. =1 if RPV is ready for pop-down =0 otherwise
IBTPE(I)	Unchanged. RPV type for RPV I: =1 for S RPV =2 for E RPV =3 for L RPV
IFUEL(I)	Unchanged. Fuel conservation status for RPV I: =1 if velocity and altitude have been altered to conserve fuel =0 otherwise
IHAND(I)	Unchanged. Handover index for RPV I. Set when operator initiates pop-up; reset when RPV is handed back to operator by terminal pilot. =1 if RPV is between handover and handback =0 otherwise
IOHND	Unchanged. Mission status index: =1 if all RPVs are through handoff and are returning =0 otherwise
IPPON(I)	Unchanged. Patch prevention index for RPV I. =1 if a new patch is not permitted =0 otherwise
IPSOH(I)	Unchanged. Waypoint index for RPV I: =1 if an S RPV has passed waypoint S =1 if an E or L RPV has passed waypoint H =2 if an S RPV has passed waypoint H =0 otherwise
ITERM(I)	Unchanged. Terminal flight index for RPV I. Set when handoff occurs; reset when terminal pilot releases control. =1 if RPV is under the control of the terminal pilot or pseudo-pilot =0 otherwise
IVPON(I)	Unchanged. Velocity change index for RPV I: =1 if a velocity change is pending for this RPV =0 otherwise
LAST (J)	New. Flag set when RPV is close to pop-up. Insures that "deferred action" limits, rather than "immediate action" limits, are used when RPV is near handoff.

NPASS (J) New. Variable that indicates which set of action limits are being employed by the operator.
=1 for "immediate action" limits
=2 for "deferred action" limits

QLOST(I) Unchanged. Flying status of RPV I:
=1 if RPV is flying
=0 otherwise

RPVNDX(I) Unchanged. Reprogramming index for RPV I:
=0.5 if RPV has been slowed down for a reroute, or if RPV is between its first major waypoint (S or H) and B
=1 if a reprogram is needed to shorten the reroute
=2 if the RPV needs to be rerouted
=3 if a malfunction has occurred
=0 otherwise

SCANTIME New. The number of milliseconds required for an operator to acquire a datum from the display screen.

TASKTIME(J) New. Variable that indicates the total elapsed time (in milliseconds) spent within task 13 before another task is scheduled.

---

exist, he begins a second pass through the flow chart. This pass begins again with a check of his terminal-area list to determine whether any pop-ups or pop-downs have become necessary. He then proceeds to check his en route/return list again using more stringent, "deferred-action" limits. Note also that a flag called LAST is set whenever an en route RPV is sufficiently close to pop-up. This flag insures that the more stringent "deferred-action" limits are used in deciding whether to patch this RPV. At any branch out of Task 13 to another task, NPASS is reset to 1 and LAST is reset to 0, so that the less-stringent "immediate-action" limits are used when Task 13 is reentered.

3.3 Model Elements

Elapsed times associated with operator tasks in the revised version of Task 13 are calculated with the aid of human performance sub-models and algorithms. Each of these sub-models and algorithms represents, with as much fidelity as is possible given our current state of understanding, the structural aspects of the perceptual, cognitive, and motor skills required in performance of the task with which it is identified. Thus, the model employed in simulating an operator's decision to correct the velocity of a vehicle in order to assure timely arrival at the hand-off waypoint assumes the existence of three distinct types of processes: (1) information acquisition, (2) numeric estimation, and (3) classification. A second example is provided by the model used in computing time taken to complete the sequence of operations involved in issuing a patch command. This model envisages six distinct operations: (1) scanning a display to obtain information, (2) pointing a light pen at the display, (3) identification of a particular function button on a keyboard, (4) depression of the button, (5) scanning the RPV track on the display, and (6) pointing the light pen at a second area of the display.

Associated with each of the perceptual, cognitive, or motor operations identified in a given model is a particular value or distribution of completion time and/or an algorithm that can be employed to generate an estimate of completion time for that operation during simulation. An estimate of the time required to complete a total task composed of these elements is achieved by summing the individual operation times. Thus, in the second of the models summarized above, an estimate of the time required by an operator to issue a patch command is achieved by adding together three scanning times, drawn independently from one distribution, to three motor performance times, computed with the aid of two additional distributions, and an algorithm for combining sample values.

The temporal distributions and algorithms employed in the current version of the simulation have different origins. One type, specified by Pritsker for the original SAINT simulation, was developed from the results of the RPV II system simulation. As noted in Section 2, this category contains models that are similar in concept to regression models, and that "describe" very accurately the results obtained in that study. The second type has been introduced by BBN on the basis of its review of the performance modelling literature. Distributions and algorithms having this latter origin have been substituted for the corresponding Pritsker formulations as part of the general effort to increase the generality of appliction of the SAINT simulation and to explore the feasibility of developing a bottom-up approach that employs existing human performance models and data.

In the remaining portion of this section, we shall present the BBN models for computation of task time and discuss the various rationales for the distributions and algorithms chosen. Information regarding the original SAINT models can be found in documentation provided by Pritsker. See Wortman et al (1976) and Duket et al (1976).

3.3.1 Structural Aspects of Human Performance Models

The structural sequences of processes assumed in models of operator tasks requiring estimation of critical mission times and interpretation of LATDEV numbers are presented in Figures 3 and 4. In these figures, the processes have been integrated into the flow of the revised simulation, as depicted in Figures 1 and 2, in order to simplify description of the use of the models for computation of elapsed task time.

The first of the models is employed where the operator must decide "Is lateral deviation number greater than the immediate action limit?". It consists of two processes, coded as e_1 and e_2 in Figure 3. The first of these is a scan of the RPV display, during which the LATDEV number of the RPV in question is acquired. The second is a classification of the acquired number into one of two categories, "greater than" or "less than or equal to" an immediate action limit (IAL) value defined by the user. If the LATDEV number falls into the former category, the operator issues an immediate patch command. If it falls into the latter category, the operator shifts his attention to the next RPV for which he is responsible.

After all vehicles on the terminal list have been exhausted and the operator has turned his attention to his en route/return

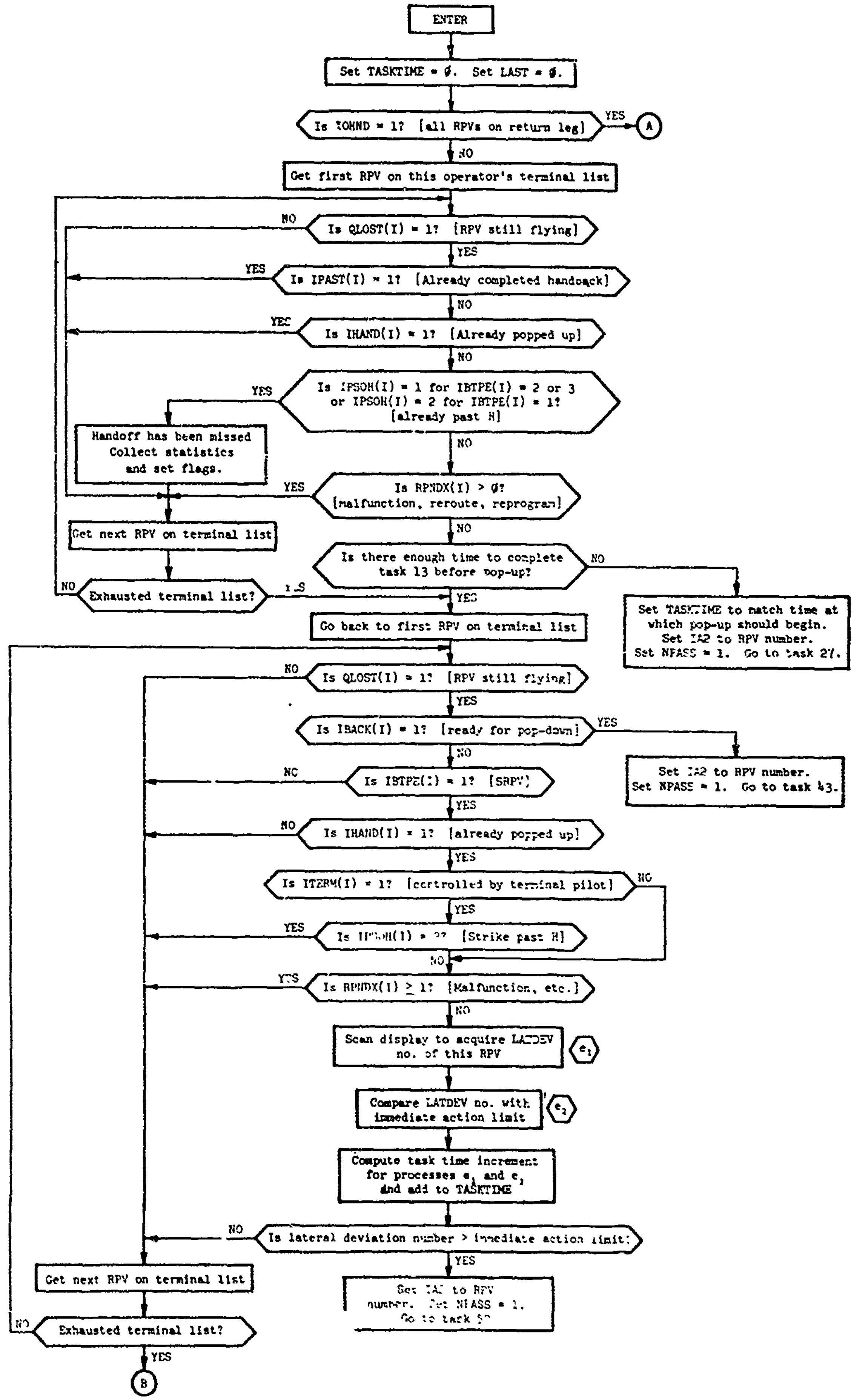

Figure 3. Task time computations for terminal area list.

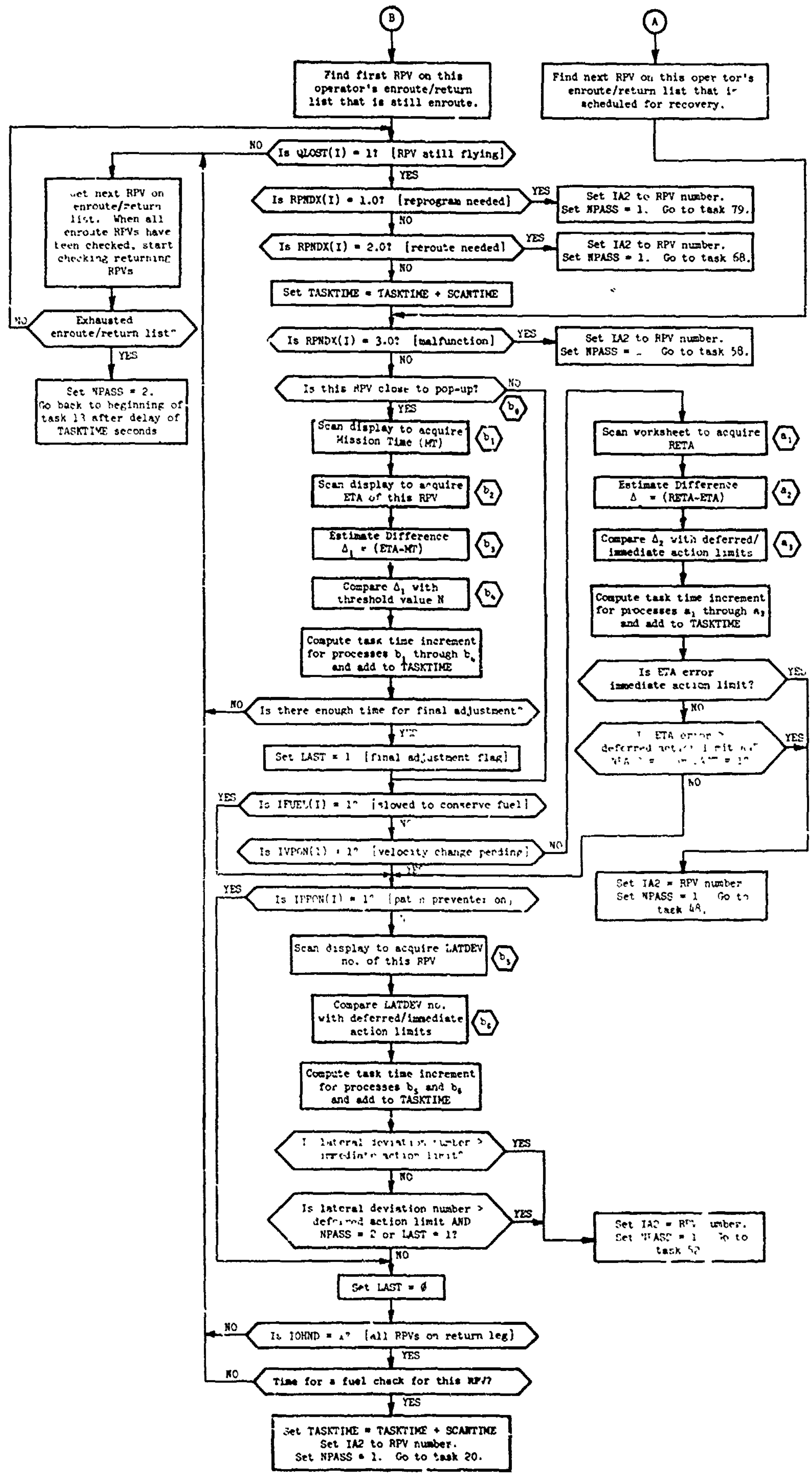

Figure 4. Task time computations for en route/return list.

list (connector B in the figures), he will proceed to determine whether the first RPV on this list is "close to pop-up." This question, coded as b_0 in Figure 4, is of little practical interest in the model, since most experienced operators can be expected to carry along in their memories a reasonably accurate picture of which RPVs are nearing pop-up. Hence, we view this question as one that "turns on" a new decision process, rather than as a decision process itself. An appropriate threshold value (say, 5 minutes) should be specified by the user. The exact value selected is not critical.

A second model is employed to answer the question, "Is there enough time for final adjustment?" It consists of four processes, labeled b_1, b_2, b_3, and b_4 in Figure 3. The first two of these are scans of the display, during which "Mission Time" (MT) and ETA are acquired. An estimate of the difference between these values is then made. In the final step, this estimate is compared with a threshold value input by the user and classified into a "greater than" or "equal to or lesser than" category, as before. If "greater than," there remains sufficient time for adjustment, and the operator proceeds to a check of RPV performance with respect to either required time of arrival (RETA) or desired flight path. If "less than or equal to," insufficient time remains for adjustment, and the operator shifts his attention to the next RPV on the list.

In the normal course of events, a velocity change commanded by an operator will not be pending (IVPON(I) will not equal 1), and the ETA check will precede a LATDEV check. The sequence of processes assumed in the model for the ETA check, labelled a_1, a_2, and a_3, is similar to that described immediately above, with the following exceptions: (1) RETA, acquired from the operator's worksheet, is employed in the estimation process in place of MT; (2) the classification process results in identification of one-out-of-three rather than one-out-of-two categories; and (3) the more-stringent "deferred action limit" (DAL) error threshold is used if the operator is performing the ETA check for the second time (NPASS = 2) or if the final adjustment flag has been set (LAST = 1). This "final adjustment" mode is intended to simulate a circumstance in which an operator adopts more stringent criteria for a particular RPV because he has determined that it is so close to pop-up that he will not have another opportunity to issue a corrective command.

If a velocity change is already pending, or if the ETA error has been found acceptable, the operator will proceed to check LATDEV unless the "patch preventer" flag is on (IPPON(I) = 1). The processes assumed here, labelled b_5 and b_6 in Figure 3, are those of scanning and classification, as before. The model differs from that employed at e_1 and e_2 in its incorporation of a trinary classification scheme similar to that contained in the model for ETA check. It also incorporates the concept of the variable error threshold employed in the ETA model.

The revised simulation also contains a model of the processes involved in the issuance of a LATDEV correction command by the operator. The structural components of this model appear in Figure 5. The elapsed time computation shown in this model should be substituted for the normal distribution employed in task 55 of the original model.

Six steps are envisioned. In the first, the operator scans the MOD Level 2 display of the RPV track and determines where to input a patch point with the light pen. After the light pen action has been taken, the operator scans the button console for the key labelled "Reconnect." After depression of the key, he again scans the display, this time in search of a region on the programmed flight path where a "reconnect" point can be input with the light pen. The final step is lightpenning the point identified.

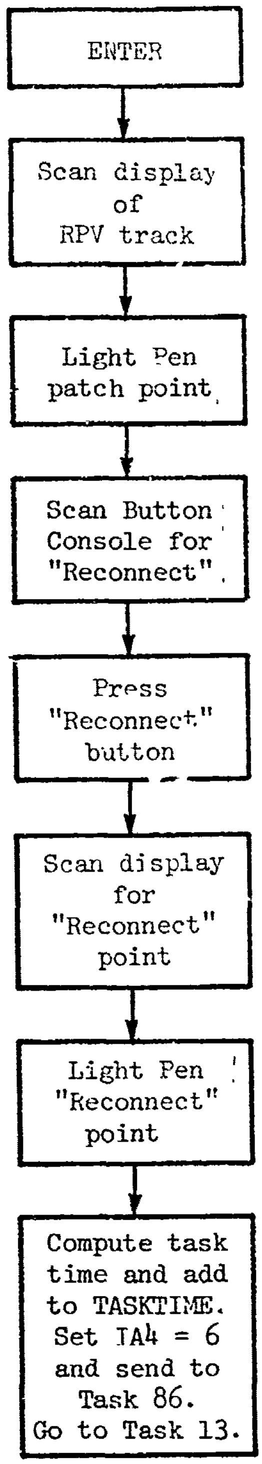

Figure 5. Task time computations for LATDEV patching

3.3.2 Computation of Elapsed Time

Estimation of the total time elapsed during operator performance of a particular RPV control task is achieved by adding together individual times associated with processes assumed in the model representing that task. The times to be employed on any given run of the simulation result either from Monte Carlo sampling from a specified distribution or from the exercise of a particular computational algorithm. Summaries of the distributions and algorithms employed are presented in Tables 2 and 3. Parameter values for the normal distributions of Table 2 were selected on the basis of a review of visual scanning, reaction time, and decision making literature during the first year of the project, and were discussed in BBN's earlier report. See Pew et al (1977, BBN Report No. 3446).

Table 2. Distributions Used in Computing Task Times

Operator process/code	Distribution type	PARAMETERS Mean (msec.)	Std. Dev.
Display scanning e_1 b_1 b_2 a_1 b_5	Normal	250	35
Button console scanning	Normal	500	150

The algorithm used to compute categorization time is well-known as "Hick's Law" (see Welford, 1968). It was originally formulated in the context of efforts to determine the functional relationship between the number of choice responses available and the minimum time required to make a choice. The algorithm has been shown to be applicable to a wide variety of situations where no motor movement is required to complete the response to a stimulus; that is, where the response consists entirely of the mental classification of an observed event.

Table 3. Algorithms Used in Computing Elapsed Time (ET)

Operator process/code	Computational algorithm	Definitions of terms
Classification	$ET=K\log_2 N$	K=simple reaction time
e_2 b_4 a_3 b_3	for $N \geq 2$	N=number of categories
Estimation of magnitude	$ET=-2.778R + 1.056$	R = (RETA-ETA)/RETA
b_3 a_2		
Movement time	$ET=a_i+b_i\log_{10}D$	a_i, b_i = constants
		$D=2A_i/W_i$
		where A_i=distance to be moved
		W_i=effective width of target area

The algorithm for computing time elapsed during an estimation process results from studies by Restle (1970) on the speed of adding and comparing numbers, and by Parkman (1971) on the judgment latencies associated with comparisons between large and small numbers. The results of these studies are reviewed in our earlier report (1977). Our algorithm assumes that estimation time (ET) is a linear function of the percentage difference (R) between the desired time of arrival (RETA) and the estimated time of arrival (ETA) for the RPV in question. (A slight variation of the model that is useful at another point in the simulation assumes, alternatively, that the percentage difference between "Mission Time" and ETA is employed in the algorithm.) At either end of the range over which this relationship is postulated to hold, estimation time is constant. For purposes of computation, the complete rule for estimation time is:

$$ET = \begin{cases} 1000 \text{ msec.} & \text{for } (RETA-ETA)/RETA < .02 \\ -2.778(RETA-ETA)/RETA+1.056 & \text{for } .02 \leq (RETA-ETA)/RETA \leq .20 \\ 500 \text{ msec.} & \text{for } (RETA-ETA)/RETA > .20 \end{cases}$$

The algorithm employed for computation of movement time is Fitts' Law, another member of the small family of well-validated empirical laws of human performance. For purposes of modelling operator movements in the current context, the following definitions of variables A_i and W_i have been adopted:

A_i: The distance traversed by right hand of operator while moving from "dwell" position to center of RPV display or to center of button console. These distances are drawn randomly from two normal distributions with mean = 8 inches, S.D. = 4 inches, and mean = 3 inches, S.D. = 1 inch, respectively.

W_i: The effective width of the display or button console target areas, defined to include 96% of all control movements. (The value of this parameter must be supplied by user.)

4. TOP-DOWN APPROACH

4.1 Introduction

4.1.1 Background

Basically, the human operator's task is to monitor the trajectories and ETAs of N vehicles, to decide if the lateral deviation or ETA error of any of these exceeds some threshold, and to correct the paths of those that deviate excessively. <*1> The top-down approach employed here uses models that have their analytical bases in control theory and in statistical estimation and decision theory. In particular, it draws heavily on the models and concepts of the Optimal Control Model of the human operator (see Baron, 1976). The modeling approach is normative, in that one determines what the human operator ought to do, given the system objectives and the operator's limitations, and this serves as a prediction of what well-trained, motivated operators will do.

It is well known that the human operator is highly adaptive, and, if motivated and given sufficient information about his performance, will attempt to change his characteristics so as to perform better. It is reasonable, therefore, to assume that a highly-trained human controller will act in a nearly optimal manner, subject to certain internal constraints that limit the range of his behavior, and also subject to the extent to which he understands the objectives of the task. This assumption is the basis of the optimal control model.

The optimal control model is a stochastic, time-domain model for the human. It includes a model for predicting the random component of human response and is not limited to stationary control situations. It is capable of treating multi-input, multi-output systems within a single conceptual framework, using state-space techniques that are naturally suited to the analysis

<*1>. In this section, the term "patch" will be used to mean either a lateral deviation patch or a velocity patch.

of complex man-machine systems. The basic model is composed of the following:

(1) an "equivalent" perceptual model that translates displayed variables into noisy, delayed perceived variables;

(2) an information-processing model consisting of an optimal estimator and a predictor that generate minimum variance estimates of the system state from the perceived data;

(3) a set of "optimal gains" chosen to minimize a quadratic cost functional (a generalization of the mean-squared error criterion that expresses task requirements); and

(4) an equivalent "motor" or output model that accounts for "bandwidth" limitations (frequently associated with neuromotor dynamics) of the human and his inability to generate noise-free control inputs.

We shall modify the optimal control model of the human operator by incorporating structures and notions that make it suitable for application to problems in which human control actions are infrequent and in which monitoring and decision-making are the operator's main activities. Thus, a combined monitoring, decision, and control model for the human operator is expected to be the end product of the top-down approach.

We plan to implement the top-down approach together with a restricted simulation of the system, DCF (Drone Control Facility), etc., to make it a self contained model (capable of functioning independently of SAINT, if necessary) so that we may utilize it to advantage to do the following:

(1) Perform a sensitivity analysis to determine the effects of parameters of the models of the system and operators on system performance, prior to or in support of SAINT implementation. The analysis would be typical of a top-down approach to prediction of system performance and it would also provide a cost effective means for establishing parameter values for the SAINT simulations.

(2) Use the sensitivity analysis to debug the top-down operator models that we will deliver to AMRL in the form of FORTRAN programs.

(3) Analyze different monitoring and control models/strategies by testing them in a restricted, self-contained simulation for preliminary screening/evaluation.

4.1.2 Description of the Model

A block diagram modeling the flow of information and the control and decisions encountered by the human operator (enroute operator) is shown in Figure 6. The DCF contains the stored flight plans that drive the N subsystems RPV_i, $i=1, 2,\ldots,N$. The true status x^i of the i-th RPV may be different from the stored flight plans due to "disturbances" w^i. The reported status y^i will be different from the true status x^i due to reporting error v_y^i. The observed status y_p^i will depend on the reported status y^i and on the "monitoring strategy" (to be discussed later on). The "information processor" processes the observed status information to produce the best estimate $\hat{x} = (\hat{x}^1, \hat{x}^2,\ldots, \hat{x}^N)$ of the true status of the N RPVs. (Note that an estimate of the state of each RPV is maintained synchronously at all times. Observation of a particular RPV improves the accuracy of the estimate of the status of that RPV while uncertainty about the status of the remaining, unobserved vehicles increases.) These best estimates are used in the "decision strategy" to arrive at a decision to (i) command a patch to one of the RPVs and/or (ii) modify the future monitoring strategy. The "patch check" block contains the "GO/NO GO" criteria to determine whether a commanded patch will "take" effect over the stored flight plan.

4.1.3 Elements of the Self-Contained Model

DCF: The stored flight plans are assumed given. (They are usually "optimal" with respect to current terrain and other information.) We will assume they can be computed using state-variable equations.

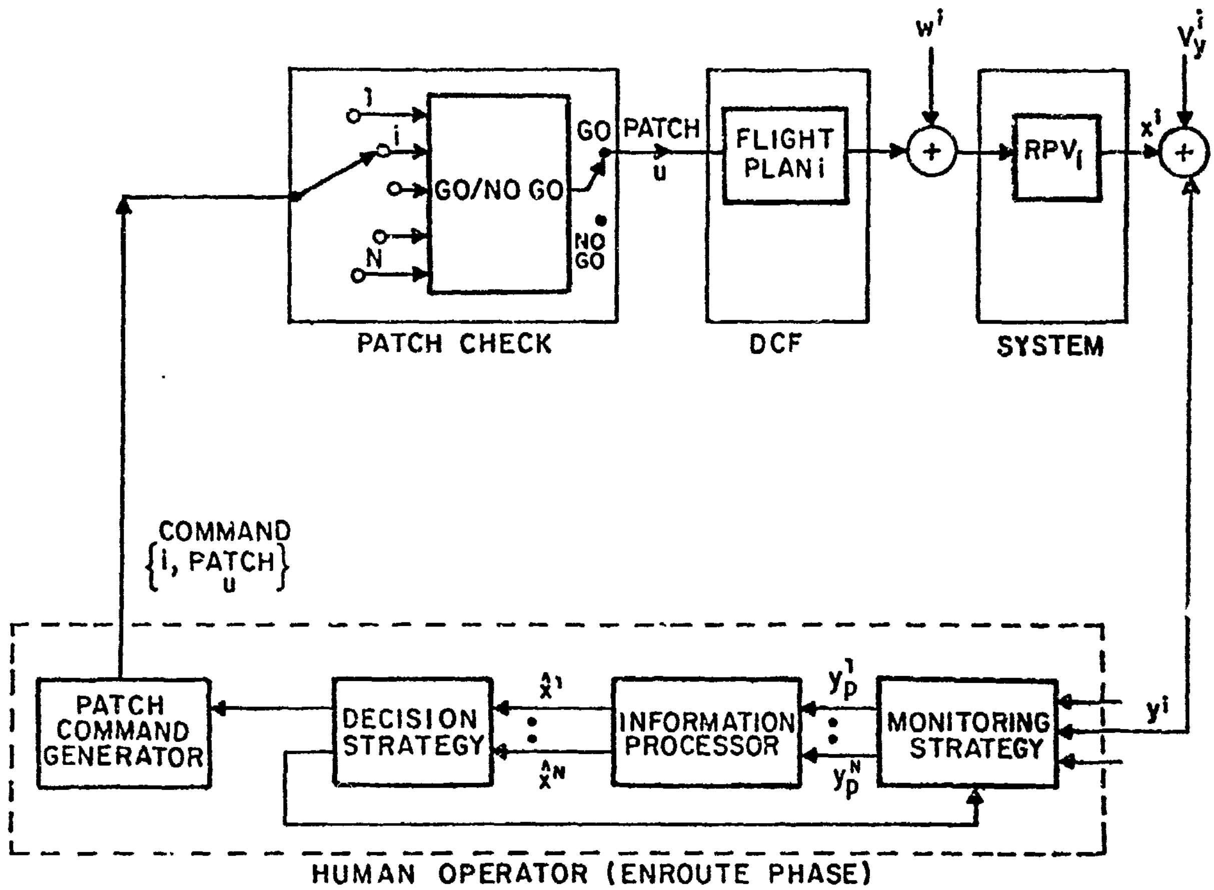

Figure 6. Block Diagram for RPV Monitoring/Control Decision Problem

System: The N RPVs undergoing monitoring/control constitute the system. A simple non-linear representation of their dynamic behavior will be assumed for this analysis. Linearization will be carried out if necessary for implementation of the model. The disturbances w^i and reporting error v_y^i will be modeled by suitable random processes. The y^i are the displayed variables corresponding to RPV_i.

Monitoring Strategy: Since the human must decide which RPV or which display to look at, he needs to develop a monitoring strategy. This is important because his estimates of the true status of each RPV (and hence his decisions) will depend upon his monitoring strategy. The following monitoring strategies seem worthy of consideration:

(i) A simple strategy involving cyclical processing of the various RPVs.

(ii) A strategy generalizing the Queueing Theory Sampling Model (Carbonell, 1966), which would minimize the total cost of not looking at a particular RPV at a given time. This strategy is mainly useful for maintaining lateral deviations within allowable limits. The costs for errors and for the different RPVs would be functions of the time-to-go and, possibly, RPV type.

(iii) A strategy aimed at minimizing total estimation error. This strategy would be consistent with monitoring for the purpose of minimizing lateral deviation errors.

Information Processor: This block models the processing that goes on in the human operator to produce the current estimate of the true RPV status from past observed status. The model anticipated for this block is the well known control-theoretic model consisting of a Kalman filter-predictor which produces the maximum-likelihood, least-squares estimate $\hat{x}$ of the true status x of all the RPVs. It also produces the variance of the error in that estimate. Given the assumptions generally made for this kind of analysis, the information processor can thus generate the conditional density of x based on the past observations y.

Decision Strategy: This block models the process of deciding which, if any, RPV to patch. We consider the decision process to be discrete (it takes 5 sec to get a new display). The cost of making a patch is the lost opportunity to monitor and/or patch other RPVs; the gain (negative cost) is the presumed reduction in error for the "patched" vehicle. The decision strategy should attempt to minimize the (expected) cost. It appears to be possible to compute such an optimizing strategy, but if this proves too difficult, a heuristic decision rule could be employed.

Patch Command Generator: This block generates the commanded patch. We anticipate a strategy based on minimizing a weighted sum of the time to return to the desired path and the total mean-square tracking error. The allowable paths would be constrained by the RPV turning radius limits. Random execution errors would be added to the commanded patch to represent human errors.

Patch Check: A GO/NO GO check will be performed on the patch using conditions on turning radius, command link status, etc.

4.2 Details of the Top-Down Model

4.2.1 System

The system under study consists of the N RPV subsystems and may be described by the state equations: <*2>

$$\dot{x} = Ax + dBu + Ew + Fz \qquad , \; x(t_o) = x_o \qquad (1)$$

where the state vector x includes the states x^i of the N-RPV subsystems. Thus, in partitioned form equation (1) appears as follows:

<*2>. For the purpose of discussion, a linear model is assumed. In actual implementation, we may use a simple non-linear model or a piecewise-linear model.

$$\begin{vmatrix} \dot{x}^1 \\ \dot{x}^2 \\ \cdot \\ \cdot \\ \dot{x}^N \end{vmatrix} = \begin{vmatrix} A^{11} & A^{12} & \cdots & A^{1N} \\ & & & A^{2N} \\ & & \cdot\cdot & \cdot \\ & & & \cdot \\ & & & A^{NN} \end{vmatrix} \begin{vmatrix} x^1 \\ x^2 \\ \cdot \\ \cdot \\ x^N \end{vmatrix} + \begin{vmatrix} d_1 I & & & \\ & d_2 I & & \\ & & \cdot & \\ & & & \cdot \\ & & & d_N I \end{vmatrix} \begin{vmatrix} B^1 \\ B^2 \\ \cdot \\ \cdot \\ B^N \end{vmatrix} u +$$

$$\begin{vmatrix} E^{11} & E^{12} & \cdots & E^{1N} \\ & E^{22} & & E^{2N} \\ & \cdot & & \cdot \\ & & \cdot & \cdot \\ & & & E^{NN} \end{vmatrix} \begin{vmatrix} W^1 \\ W^2 \\ \cdot \\ \cdot \\ W^N \end{vmatrix} + \begin{vmatrix} F^{11} & F^{12} & \cdots & F^{1N} \\ & F^{22} & & F^{2N} \\ \cdot & \cdot & \cdot & \cdot \\ & \cdot & \cdot & \cdot \\ & & & F^{NN} \end{vmatrix} \begin{vmatrix} Z^1 \\ Z^2 \\ \cdot \\ \cdot \\ Z^N \end{vmatrix} \qquad (2)$$

For the system under study, the following observations hold:

A1: Only one of the N-RPV subsystems may be controlled by the patch-control u at any given time. A decision to control the i-th RPV subsystem then implies the following conditions on the decision variables:

$$d_i = 1 \quad , \; d_j = 0 \qquad j \neq i \qquad (3)$$

A2: The N-RPV subsystems are decoupled (except for the interdependence of the decision variables via (3)), that is,

$$A^{ij} = 0 \;\; , \;\; E^{ij} = 0 \; , \; F^{ij} = 0 \qquad i \neq j \qquad (4)$$

The N-RPV subsystems may thus be described by

$$\dot{x}^i = A^{ii} x^i + d_i B^i u + E^{ii} w^i + F^{ii} z^i \; , \quad x^i(t_o) = x_o^i \quad (5a)$$

$$d_i = 0 \text{ or } 1 \qquad (5b)$$

$$\sum_{i=1}^{N} d_i = 1 \tag{5c}$$

4.2.2 Flight Plan (DCF)

When there is no disturbance w^i and no (patch) control u then the N-RPV subsystems follow the flight plan $\bar{x}^i$

$$\dot{\bar{x}}^i = A^{ii}\bar{x}^i + F^{ii}z^i \quad , \quad \bar{x}^i(t_O) = \bar{x}_O^i \tag{6}$$

Flight plans made up of straight lines are easily generated using a piecewise constant time function for z^i and $\bar{x}_\emptyset^i$ as the launch point.

4.2.3 Patching

Any disturbances w^i causes the i-th RPV to deviate from its flight plan. With $e^i = x^i - \bar{x}^i$ it follows from (5) and (6) that

$$\dot{e}^i = A^{ii} e^i + d_i B^i u + E^{ii} w^i \quad , \quad e^i(t_o) = x_o^i - \bar{x}_o^i \tag{7a}$$

$$d_i = 0 \text{ or } 1 \tag{7b}$$

$$\sum_{i=1}^{N} d_i = 1 \text{ or } 0 \tag{7c}$$

It is the purpose of the (patch) control u to correct any such deviation. Since w^i is an unknown random disturbance and d^i is nonzero for at most a single RPV subsystem, it is not possible to maintain $e^i = \emptyset$ for all i. We shall resolve the patching problem via the following three sub-problems:

(i) Patching decision - which RPV to patch?

(ii) Patch command computation and generation (iii) GO/NO GO check on the patch (e.g., observe minimum turn-radius condition on RPV).

4.2.3.1 Patching Decision

A patching decision consists of deciding which RPV subsystem is to be patched, if any. At most one of the RPVs may be patched at a given time. One idea of patching is to reduce deviations from the flight path to below some threshold values. Some facts to note are:

(i) Cross-track error of less than 250' is desired for type-S RPVs

(ii) Terminal-phase control not possible if cross track error exceeds 1500'

We assume a normative model, in which the operator attempts to optimize some (subjective) measure of performance via a patching decision. For this purpose, we consider two alternative cost functions to arrive at a patching decision:

Piecewise constant cost function

$$C(e^i) = \bar{C}^i \qquad \text{if} \quad e^i \in e^i_T \text{, a threshold set}$$

$$C(e^i) = C^i \qquad \text{if} \quad e^i \notin e^i_T \text{.}$$

Quadratic Cost function

$$C(e^i) = e^{i'} K e^i$$

The choice of e^i_T and K will be made based on facts of the type (i) and (ii) noted above. The costs C^i, $\bar{C}^i$, $C(e^i)$ will be chosen to be functions of mission time to reflect the importance of ETA. As mission time gets closer to ETA for RPV-i, C^i will be made larger and/or e^i_T will be shrunk to reflect "urgency". The optimal patch decision will be chosen to minimize the expected cost using subjective probabilities computed with the help of the information processor. The details are in Appendix B.

4.2.3.2 Patch Control Computation and Generation

Once a decision is made to patch a particular RPV-subsystem, it is necessary to compute and execute the patch control. The purpose of a patch control is to guide the aircraft from its initial location and heading to intercept and fly along the planned flight path. Various criteria may be considered to compute the optimal patch control. We shall consider a strategy that minimizes the time to return to the planned flight path. The details are in Appendix A (see also Erzberger, 1971).

4.2.3.3 GO/NO GO Check

This consists of checking minimum turn-radius violation, command-link status, fuel availability, time-sequencing, etc. The Go/No Go check is independent of the operator and will be implemented in a manner similar to that in the Bottom-up approach.

4.3 Implementation of the Top-down Approach

The combined monitoring, decision, and control problem that arises in the top-down approach to modeling the enroute operator will be implemented in FORTRAN. The program will have a modular structure to facilitate ease of adding further modules to include alternative monitoring, control, and decision strategies that may appear promising at a future date. Moreover, the modules comprising the human operator part alone may then be used in the SAINT simulation to produce the patch decisions and patches based on the SAINT displayed outputs.

To accomodate the random aspects of the problem, the program will basically have a Monte-Carlo simulation character. It will produce as outputs the "true" time-histories of the RPV flights, the sequence of monitoring and patching decisions made, and the resulting performance.

The important aspects of the simulation program implementing the top-down model for the combined monitoring, decision, and control problem are shown in the flow diagram in Figure 8. There are, as indicated, nine major modules in the program. Modules 4, 6 and 7 are of special interest because they do not arise in the usual manual control models. The theory behind these modules is developed in Appendices A and B. As indicated in Appendix A, the patch command generator would involve a non-linear control law.

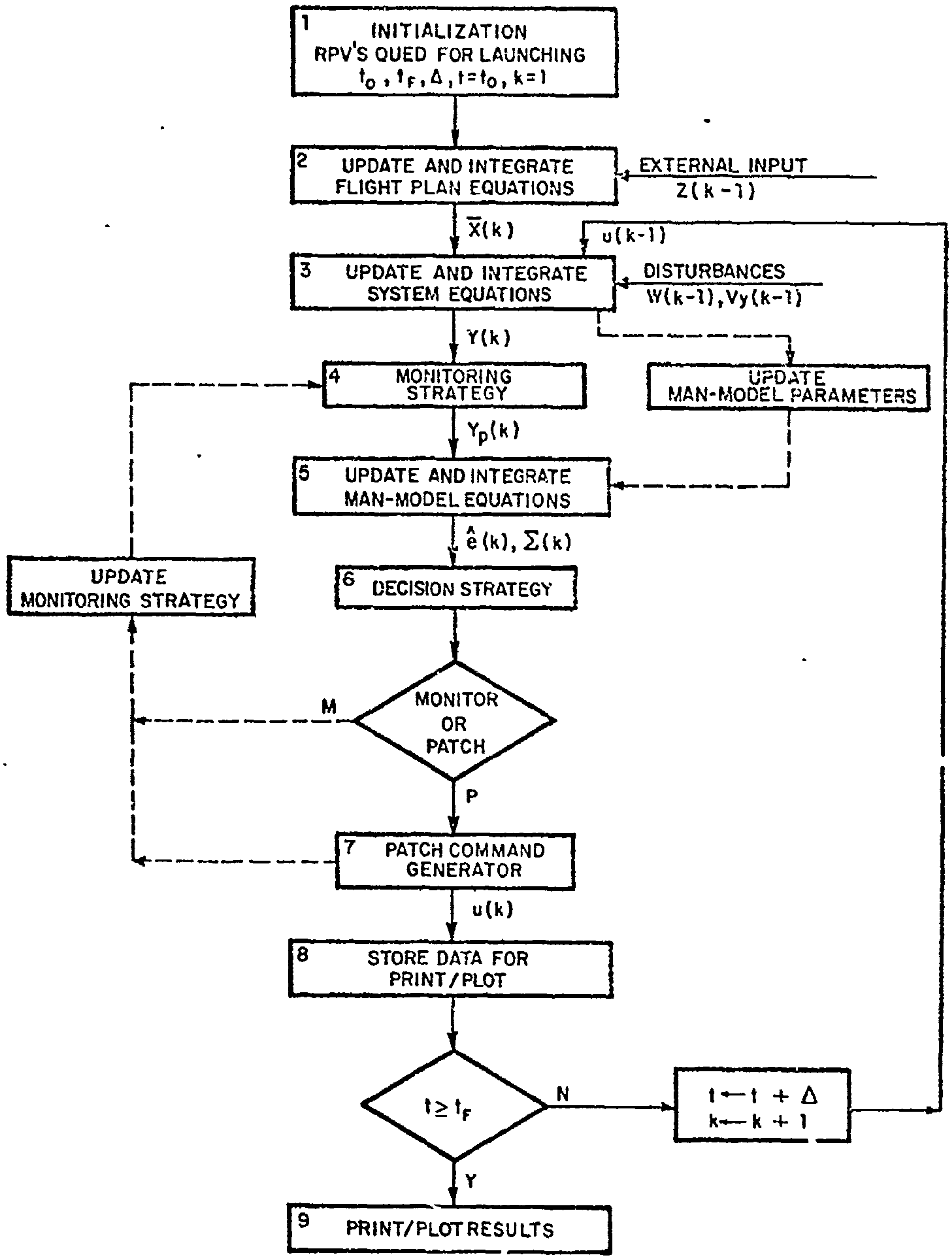

Figure 7. Flow Diagram for the Top-down Model Implementation

5. PROBLEMS AND ISSUES ASSOCIATED WITH MODEL DEVELOPMENT

At this writing, preliminary runs of the bottom-up model at AMRL have not yet been completed. One consequence of this is that it is not possible to evaluate the success of our attempt to reconfigure the SAINT/RPV II simulation model and to introduce human performance sub-models. Another is that it is not possible to classify issues that have arisen during the effort in terms of their generality to other modelling environments.

Despite these limitations, however, our review of the modelling literature during the first year of the project and our actual modelling efforts to date have highlighted a number of problems that may be of sufficient interest to warrant summary here. Because the bottom-up and top-down models have been developed on somewhat different time tables, we have elected to treat the two separately in the section that follows.

5.1 Problems Associated with Properties of the RPV Control Task

In essence, the RPV control task is one in which long periods of flight following and resource monitoring are occasionally interrupted by short sequences of corrective actions requiring a modicum of skilled motor performance. Infrequently, there are also instances in which members of a control team must communicate with each other to effect orderly exchange of aircraft at transition points in a mission.

In the prototype RPV control system developed at AMRL, the pace of monitoring, corrective and communication actions is determined by the basic five-second update rate of the information displays. This quantization may result, on occasion, in an operator's decision to defer until the start of the ensuing "frame" an action that he might take immediately if he were presented instead with continuously updated information.

From the point of view of bottom-up modelling, three significant problems emerge in connection with the task as depicted in Section 2.1 of this report and summarized above. All result from lack of availability of models appropriate to the subtask structure as portrayed by classical task analysis. The first of these problems has to do with the lack of balance between explicit (observable) and implicit (unobservable) behavior associated with the control task itself; the second, with the "team" aspect of performance; the third, with the communication requirements.

5.1.1 Lack of Balance Between Implicit and Explicit Processes

If, as here, a goal of the bottom-up approach is to develop a model that is faithful both to the "conduct" of the control process and to the output(s) of the process, one must, by definition, have available valid sub-models that can be mapped on to the set of real activities identified in a prior task analysis. Finding and adapting such models appears to be relatively easy when the control activities are explicit and result in defineable outcomes. Our straightforward modification of the Fitts algorithm for use in SAINT serves as an ideal example. Where, however, a significant portion of the control task is composed of largely implicit activities that result in only occasional observable output, such as "monitoring," "evaluation" and "estimation," the problem of finding previously validated quantitative models appropriate to the control context may be very difficult. Our use of an unvalidated combination of the Restle (1970) and Parkman (1971) data to model controller estimation of time-to-hand-off provides an example of one possible response to deficiencies in the literature. Because of the possibility of propagated errors, however, there is the possibility that application of more than one unvalidated model at a time to a sequence of implicit activities may complicate significantly the task of properly apportioning variance between model components during later sensitivity analyses. Consequently, we do not view such application as a general solution to the problem.

We believe that there are three alternatives open to model developers in these circumstances:

1) Relax the criterion that the model shall faithfully represent process as well as output. Selection of this alternative may force departures from prior task analyses that highlighted the extent of the cognitive task loading to begin with.

2) Attempt to retain the form of the process at a descriptive level but substitute fixed response thresholds that will yield "empirically valid" output at the end of each process link associated with totally implicit behavior. Selection of this alternative may result in a model that frequently replicates actual performance but that may be insensitive to changes in implicit decision criteria. Further, it seems clear that, by adopting this alternative, one may merely displace the ambiguity from one level bottom-up formulation to some lower level and not solve the problem in principle.

3) Depart from a bottom-up approach and the requirement to replicate "process," and pursue the more limited goal of formulating a model that successfully reproduces the final outputs of a sequence of behaviors. This, of course, may lead to a top-down approach or to some intermediate level in which non-cognitive sub-task models are combined with top-down models in order to achieve a desired level of model performance.

Our bottom-up formulation of operator performance generally reflects adoption of the second of these alternatives. We have chosen to represent the largely unobservable task of monitoring as a series of discrete processes of display scanning, threshold comparison and time estimation. Some flexibility is incorporated into an otherwise lock-step procedure by the use of immediate and deferred action limits. Further, the scanning behavior has been modified to make it more representative of the way in which a human operator would be likely to monitor his display. The result, however, remains a compromise on the process as identified in the original task analysis and may not capture at all well the dynamic aspects of the control task.

5.1.2 Team Performance

Although the frequency of occasions on which an RPV control team must function as a collective is low, and the amount of time elapsed during these occasions small when compared with total task time, the success of the mission depends critically on the extent to which team members can cooperate when required. Given this characteristic, an important issue one might want to address via simulation during initial design of control procedures is how performance varies as a function of such factors as team morale, cohesiveness and training method. As noted in our earlier report, however, few models of group performance exist in sufficiently quantitative form to be applicable to the control context. One possible exception to this generalization is the model framework utilized by Siegel and Wolf (1969) in their work on crew performance. Unfortunately, the Siegel and Wolf model requires as input much information derived from prior empirical study of operator performance and, as a result, probably cannot be used successfully here.

In the current version of the bottom-up model, the team aspects of RPV control remain unmodelled. As discussed in Section 3.2, we have assumed that operators are identical in their behavioral characteristics and, further, that the simulated performance of any one of the operators at any given time is based entirely on the types of RPVs that operator must control. As such, the performance of any simulated team will be exactly like the performance of any other simulated team so long as RPV parameters are held constant.

5.1.3 Communication Requirements

As with other aspects of their interaction, the quality of verbal communication between controllers during critical phases of the mission (e.g., hand-off and hand-back) contributes to the success of the team effort. However, for the same reason that limits a general attack on other parameters of team enterprise at this time, viz., unavailability of models, operator communications are not modelled in the current formulation. Instead, we assume that all required information is correctly exchanged and that all departures from desired performance are due to individual, as opposed to team, control failures.

Although we have not yet attempted it, we expect that the task of developing a model of communication suitable for use in the bottom-up approach will prove to be much less difficult than that associated with team variables discussed in Section 5.1.2. This expectation results from the fact that (1) circumstances in which communications must occur are well-defined and can be referenced to specific points along the flight path or to specific temporal intervals during the mission, and (2) the nature of the information that must be transmitted from one operator to another can be rather precisely characterized. As such, the communications are very likely to be amenable to modelling via an information theoretic framework.

5.2 Problems Associated with the Existing SAINT/RPV Model

The design of the existing SAINT/RPV II simulation model has created some major practical impediments to the smooth introduction of sub-models and to the conduct of sensitivity analyses. The most significant of these has to do with the very high degree of interconnectedness between the complex of moderator (MODRF), and user (USERF) functions and the main program. Our understanding of the rationale underlying the incorporation of the MODRFs and USERFs is that they provide ideal mechanisms for updating program variables and operator status, and for coordinating state variable and task oriented components of the RPV model. The overall updating and coordination objectives are achieved in the model, but there is a lack of consistency as to where and in what way these objectives are met on an individual basis. The result is that much of the modularity that might otherwise be gained with the use of separate MODRFs and USERFs is lost, and it becomes extremely difficult to introduce new sub-models without adversely affecting the integrity of the rest of the existing framework.

The most common manifestation of these difficulties arises from the use of multiple flags that are set and reset by various MODRFs and USERFs as different events occur. With some effort, one can search through the code and discover the points at which these settings take place, and can insure that revised versions of subtask models handle the flags in similar ways. One often discovers, however, that other modules begin behaving differently even though the greatest care has been taken; it turns out that other USERFs connected with other subtasks have also been comparing the flag settings and taking various actions based thereon. As a result, what at first glance appear to be completely independent modules turn out to be quite interrelated. Our experience so far suggests that successful modifications to the subtask structure of SAINT/RPV II cannot be successfully accomplished without an inordinately large investment in time to debug the resulting model. It does seem clear, however, that once this investment is made, further modifications can proceed without undue delay.

5.3 Issues in the Top-Down Modelling Approach

5.3.1 Model Validation

We have explored this issue on a preliminary level and validated the major conceptual process of the top-down approach. A qualitative model validation was obtained by exercising our DEMON model on a simple RPV-example and obtaining reasonable results. A more detailed RPV model is being developed and studied with a view to providing detailed model identification and quantitative model validation against "data."

5.3.2 Discrete Versus Continuous Tasks

One of the major issues confronting the use of control theoretic models as the basis for the top-down approach is how to treat asynchronous, discrete tasks in what is essentially a continuous set-up. We are resolving this issue by embedding the discrete monitoring tasks and continuous control tasks (which occupy a "chunk of time" during the mission) in an overall combined decision, monitoring and control task which is continuous and occupies the entire mission duration. The parameters of the decision task determine at what instants of time a particular discrete monitoring or continuous control task is released.

5.3.3 (Under) Determination of Multi-Parameter Models

The parameters of the DEMON model for the enroute operator include the standard parameters of the OCM describing human processing limitations (i.e., the time delay and observation noise). In addition, two monitoring parameters and two patching parameters are needed. They are:

i) monitoring cost ii) monitoring threshold

iii) patching cost iv) patching threshold

It is clear that these do not represent a unique set of parameters for describing the operator. On the other hand, the parameters of the model are not great in number and it may be possible to specify or determine unique values for them in a consistent and logical manner.

REFERENCES

Baron, S., "A Model for Human Control and Monitoring Based on Modern Control Theory," **Journal of Cybernetics and Information Sciences**, Vol. 4, No. 1, Spring 1976

Baron, S. and W. Levison, "An Optimal Control Methodology for Analyzing the Effects of Display Parameters on Performance and Workload in Manual Flight Control, "**IEEE Trans. on Systems Man and Cybernetics**, Vol. SMC-5, No. 4, July 1975.

Bryson, A. E. and Y. C. Ho, "Applied Optimal Control", Hemisphere, 1969

Baron, S. and D. L. Kleinman, "The Human as an Optimal Controller and Information Processor," **IEEE Trans. on Man Machine Systems**, MMS-10, No. 1, March 1969.

Baron, S., D. L. Kleinman, et al., "Application of Optimal Control Theory to the Prediction of Human Performance in a Complex Task," Wright Patterson Air Force Base, Ohio, AFFDL-TR-69-81, March 1970.

Carbonell, J. R., "A Queueing Model of Many-Instrument Visual Sampling," IEEE **Transactions on Human Factors in Electronics**, Vol. HFE-7, No. 4, December 1966, pp. 157-164.

Erzberger, H. and H. Q. Lee, "Optimum Horizontal Guidance Techniques for Aircraft," **J. Aircraft**, Vol. 8, No. 2, February 1971, pp. 95-101.

Kleinman, D. L. and S. Baron, "A Control Theoretic Model for Piloted Approach to Landing," **Automatica**, Vol. 9, 1973, pp. 339-347.

Kleinman, D. L., S. Baron and W. H. Levison, "An Optimal-Control Model of Human Response, Part 1: Theory and Validation," **Automatica**, Vol. 6, 1970, pp. 357-369.

Pew, R. W., S. Baron, C. E. Feehrer, and D. C. Miller, "Critical Review and Analysis of Performance Models Applicable to Man-Machine Systems Evaluation," BBN Report No. 3446, March 1977

APPENDIX A: TOP-DOWN CONTROL STRATEGY

A.1 System Dynamics and Patch Computation

In Section 2, the N-RPV system dynamics were considered in general terms. Here, we shall consider some simple models for the RPV-subsystem dynamics. Only projected motion in the horizontal plane is considered. Pop-up and pop-down and consideration of fuel constraints will be taken up at a later stage during implementation of the top-down model.

Once a decision is made to patch a particular RPV-subsystem, it is necessary to compute and execute the patch control. The purpose of a patch control is to guide the aircraft from its initial location and heading to intercept and fly along the planned flight path. Various criteria may be considered to compute the optimal patch control. The normalized equations of motion derived in Erzberger (1971) based on a set of simplifying assumptions are:

$$\dot{x} = v_x \tag{1}$$

$$\dot{y} = v_y \tag{2}$$

$$\dot{v}_x = u\, v_y \tag{3}$$

$$\dot{v}_y = -u\, v_x \tag{4}$$

where (x,y) denote the normalized position in the x-y plane, (v_x, v_y) denote the normalized velocity components in the (x,y) directions, and u is the normalized horizontal force on the aircraft due to its bank angle. It is required that the velocity be constant and normalized such that

$$v_x^2 + v_y^2 = 1.$$

From (3) and (4) it follows that

$$v_x \dot{v}_x + v_y \dot{v}_y = 0 \text{ or } v_x^2 + v_y^2 = \text{const.}$$

for all t. Thus, if

$$v_x^2(0) + v_y^2(0) = 1$$

then (3) and (4) satisfy the requirement that

$$v_x^2 + v_y^2 = 1 \text{ for all } t.$$

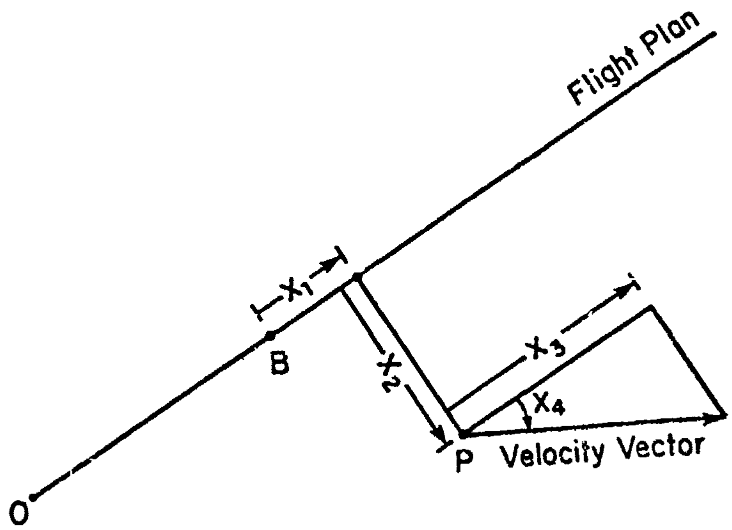

Figure 8. Choice of Co-ordinates for System Equation

We shall re-write the system equation using (see Figure 8)

x_1 = ground-speed error,

x_2 = cross-track error,

x_3 = velocity component along track,

x_4 = heading relative to track:

$$\dot{x}_1 = \cos x_4 - 1 \quad , \; x_1(0) \text{ given}, \quad x_1(T) \text{ free} \tag{5}$$

$$\dot{x}_2 = \sin x_4 \quad , \; x_2(0) \text{ given}, \quad x_2(T) = 0 \tag{6}$$

$$\dot{x}_3 = u \sin x_4 \quad , \; x_3(0) \text{ given}, \quad x_3(T) = 1 \tag{7}$$

$$\dot{x}_4 = -u \quad , \; x_4(0) \text{ given}, \quad x_4(T) = 0 \tag{8}$$

$$T \quad \text{free} \tag{9}$$

$$x_4^2(0) + x_3^2(0) = 1 \tag{10}$$

The optimal patch control u will be computed by minimizing the performance criterion,

$$J = 1/2K_1X_1^2(T) + 1/2 \quad K_2 \int_0^T x_2^2 \, dt + K_3 \int_0^T 1 \, dt \tag{11}$$

which is a weighted sum of the square of the ground speed error, integral square of the cross-track deviation, and time to return to the planned flight path.

If we chose the weights to be $K_1=\emptyset=K_2$ and $K_3=1$ then J is the time to return to the flight path. We shall only solve this special problem of minimum time to return to the flight path at this time. The general problem will be considered at a later date, if necessary for a proper resolution of velocity patch, fuel constraints, etc.

A.2 Minimum Time Patch Strategy

The Hamiltonian in this case is

$$H = 1 + \lambda_1(\cos x_4 - 1) + \lambda_2 \sin x_4 + u\,(\lambda_3 \sin x_4 - \lambda_4) \tag{12}$$

The necessary conditions for minimum time yield

$$\dot{\lambda}_1 = 0 \quad , \lambda_1(T) = 0 \tag{13}$$

$$\dot{\lambda}_2 = 0 \quad , \lambda_2(T) \quad \text{free} \tag{14}$$

$$\dot{\lambda}_3 = 0 \quad , \lambda_3(T) \quad \text{free} \tag{15}$$

$$\dot{\lambda}_4 = \lambda_1 \sin x_4 - \lambda_2 \cos x_4 - u\,\lambda_3 \cos x_4 \ , \ \lambda_4(T) \ \text{free} \tag{16}$$

Since H is independent of t, H=const. $=\emptyset$ by transversality condition. Since H is linear in u, and $|u| \leq 1$, control is Bang-Bang except for possible singular arcs. The optimal minimum time patch control is

$$u = \text{sgn}\ S \tag{17}$$

where the switching function

$$S = \lambda_3 \sin x_4 - \lambda_4$$

To compute the singular control we insist that

$$S \equiv 0 \tag{18}$$

$$\dot{S} = \lambda_1 \sin x_4 + \lambda_2 \cos x_4 \equiv 0 \tag{19}$$

$$\ddot{S} = -u\,(\lambda_1 \cos x_4 - \lambda_2 \sin x_4) \equiv 0 \tag{20}$$

From (2Ø) either u=Ø or $\lambda_1 \cos x_4 - \lambda_2 \sin x_4 = 0$. But in view of (19) the latter condition would require $\lambda_1 = 0 = \lambda_2$ and hence H=1 which is impossible. Hence $u \equiv 0$ is the singular control. Along singular arcs then

$$-1 = \lambda_2 \sin x_4 + \lambda_1 (\cos x_4 - 1)$$

$$0 = \lambda_1 \sin x_4 + \lambda_2 \cos x_4$$

That is,

$$\lambda_1 = \frac{\cos x_4}{(\cos x_4 - \cos 2 x_4)}$$

Since $\lambda_1 = 0$ from (13) it follows that on singular arcs $x_4 = \pm \pi/2$.

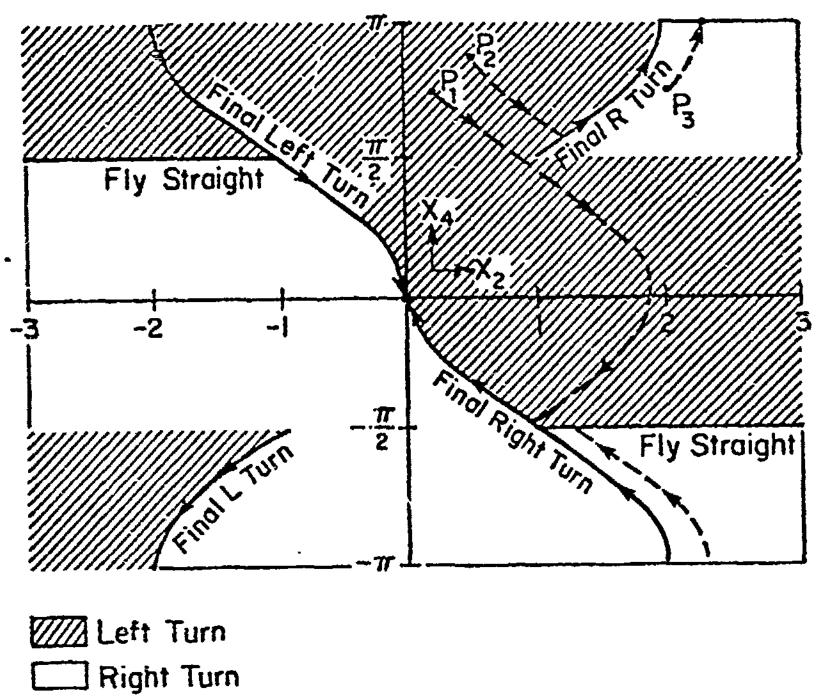

Figure 9. Minimum Time Patch Control Strategy

It is straightforward to compute the minimum-time patching strategy, and the result is indicated in Figure 9. For example, all points in state space that can be brought to the planned flight path using a single left turn u=1 are characterized by the equation $x_2(0) = \cos x_4(0)-1$ which is obtained from

$$\dot{x}_4 = -1 \ , \ x_4(T) = 0$$

$$\dot{x}_2 = \sin x_4 \ , \ x_2\ (T) = 0$$

The minimum time required for the patch will be checked against the scheduled pop-up/pop-down times for the given RPV to determine if the computed patch should be executed. Velocity patches to correct for ETA errors with due regard to fuel constraints may be included by a simple extension of the above problem (for example, append to the minimum time patch a velocity patch to minimize ETA errors). This will be done at a later stage during the implementation of the model.

The operator does not observe the states x directly, and will base his control actions instead on the best estimates of these states available to him based on all his observations. This is in line with the "separation principle" that separates estimation and control (see Bryson and Ho, 1969).

APPENDIX B: TOP-DOWN DECISION STRATEGY

B.1 Monitoring and Patching Decision

As stated in section 4.1.2, the information processor produces the current estimate of the true status x of all the RPVs at any time. It also produces the variance of the error in that estimate. Thus, the information available for making monitoring and patching decisions consists of the posterior distribution of e^i conditioned on all observations based on past monitoring and patching decisions and control. Under the usual assumptions, this posterior distribution for e^i is $N(\hat{e}^i, \Sigma^{ii})$.

Let e_T^i denote a threshold set associated with the i-th RPV, that is, $e^i \epsilon e_T^i$ is a desirable condition. Let H^i denote the hypothesis that $e^i \notin e_T^i$ and P^i be the probability that H^i is true. P^i is easily calculated using the available information on the posterior distribution of e^i:

$$P^i = 1 - \int_{e_T^i} N(\hat{e}^i, \Sigma^{ii})\, de^i \tag{1}$$

Monitoring the i-th RPV results in a tighter distribution for e^i around its mean $\hat{e}^i$ because it reduces the uncertainty ^{ii} associated with $\hat{e}^i$. This makes P^i closer to 0 or 1 and thus helps in deciding accurately if H^i is true or not. Patching the i-th RPV requires monitoring as well. The effects of patching are: first, to correct the error e^i which might have 'wandered off' from zero due to disturbances, by assuring that $\hat{e}^i \epsilon e_T^i$; and second, to provide a tighter distribution of e^i around its mean $\hat{e}^i$. The cumulative effect of patching is to make P^i closer to zero.

To formulate and solve the combined monitoring and patching decision problem, we shall assume that C_i is the cost if H^i is true. Recall that H^i has a (subjective) probability P^i of being true. Just as H_i, P^i, C_i were defined in relation to the set e_T^i, let $\bar{H}_i$, $\bar{P}^i$, $\bar{C}_i$ be defined in relation to the set $\bar{e}_T^i$, the complement of e_T^i. We shall use minimum expected cost EC(d*) as the criterion for selecting the best monitoring and patching decision d*.

Let d_{ij} denote a decision to monitor RPV-i and patch RPV-j in the combined monitoring and patching decision problem. Since a patch can be done only on a monitored RPV, there are only 2N+1 available decisions. They are:

(i) Do nothing decision $d_{\emptyset\emptyset}$, that is, monitor no RPV and patch no RPV.

(ii) N pure monitoring (no patching) decisions $d_{j\emptyset}$, j=1,2,..,N.

(iii) N patching (and monitoring) decisions d_{jj}, j=1,2,...,N.

Let P_{ijk} denote the probability that the hypothesis H^i is true (that is, RPV-i is outside the threshold set indicated by $e^i \notin e^i_T$) when the decision is d_{jk}. Because the RPV subsystems are non-interactive, it follows that the probabilities associated with RPV-i when some other RPV is monitored and/or patched is same as that associated with RPV-i when no RPV is monitored. That is,

$$P_{i\emptyset\emptyset} = P_{ijk} \quad \text{any } j \neq i,\ i = 1, 2, \ldots N;\ k = j \text{ or } \emptyset$$

Thus, there are only 3N distinct probabilities to be computed

(i) N probabilities $P_{i\emptyset\emptyset}$ associated with do-nothing decision $d_{\emptyset\emptyset}$

(ii) N probabilities $P_{ii\emptyset}$ associated with pure monitoring decision $d_{i\emptyset}$

(iii)N probabilities P_{iii} associated with patching decision d_{ii}

Let $(PP)_i$ denote the probability that the patch decision d_{ii} "takes", that is, results in e^i e^i_T, and let T_{ij} denote the cost of implementing decision d_{ij}. The costs T_{ij} will be chosen to be functions of mission time to reflect the importance of ETA. As mission time gets closer to ETA for RPV-i, T_{ij} will be made larger and/or e^i_T will be shrunk to reflect "urgency" in the same spirit as the "immediate" and "deferred" action limits introduced in the bottom-up approach.

The combined monitoring and patching decision problem may be described in terms of a decision-tree diagram as shown in Figure 10. The actual cost of a particular decision depends on the path chosen to traverse the tree from level 1 to level 5. The exact

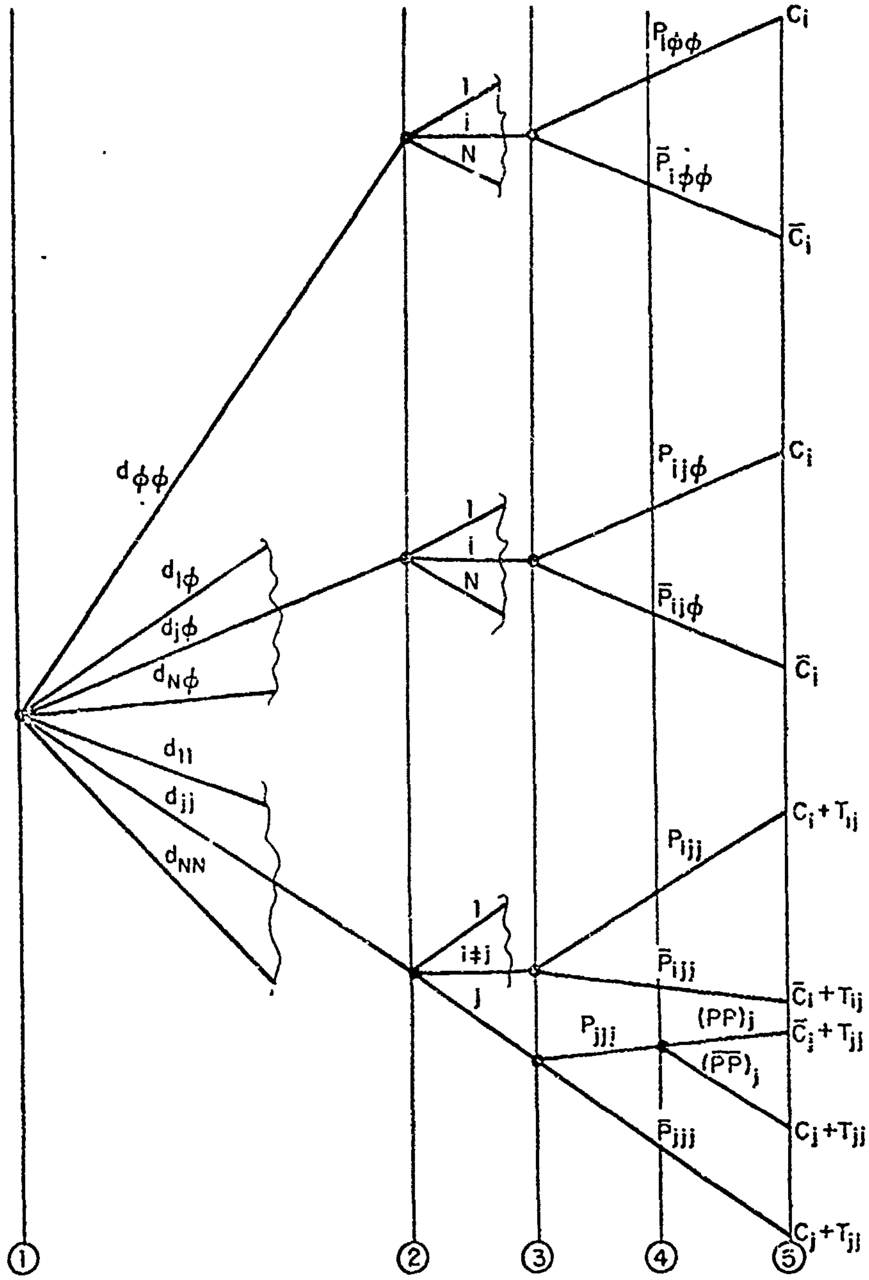

Figure 10. Decision Tree Diagram for Combined Monitoring and Patching

path from level 1 to level 5 for the N-RPVs are determined both by the decision maker (the human operator) and by Nature (the random elements in the problem) . Since a decision has to be made at level 1 **before** Nature has taken its course at the monitoring level 3 and at the patching level 4 , the decision maker can only evaluate his 2N+1 alternative decisions in terms of their expected costs. This he can do as follows: The expected cost of the do-nothing decision $d_{\emptyset\emptyset}$ is

$$EC(d_{\emptyset\emptyset}) = \sum_{i=1}^{N} (C_i P_{i\emptyset\emptyset} + \bar{C}_i \bar{P}_{i\emptyset\emptyset}) \tag{2}$$

Expected cost of pure monitoring decision $d_{j\emptyset}$ is

$$EC(d_{j\emptyset}) = \sum_{i=1}^{N} (C_i P_{ij\emptyset} + \bar{C}_i \bar{P}_{ij\emptyset}) + T_{j\emptyset} \tag{3}$$

$$= EC(d_{\emptyset\emptyset}) - (C_j P_{j\emptyset\emptyset} + \bar{C}_j \bar{P}_{j\emptyset\emptyset}) + (C_j P_{jj\emptyset} + \bar{C}_j \bar{P}_{jj\emptyset}) + T_{j\emptyset} \tag{4}$$

Expected cost of a patching decision d_{jj} is,

$$EC(d_{jj}) = \sum_{\substack{i=1 \\ i\neq j}}^{N} (C_i P_{ijj} + \bar{C}_i \bar{P}_{ijj}) + [P_{jjj}\{C_j (\overline{PP})_j + \bar{C}_j (PP)_j\} + \bar{P}_{jjj} \bar{C}_j + T_{jj}]$$

$$= EC(d_{\emptyset\emptyset}) - (C_j P_{j\emptyset\emptyset} + \bar{C}_j \bar{P}_{j\emptyset\emptyset}) + (C_j P_{jjj} + \bar{C}_j \bar{P}_{jjj}) - \{(PP)_j P_{jjj} (C_j - \bar{C}_j) - T_{jj}\} \tag{5}$$

The optimal decision d* is the one which results in maximum opportunity gain, that is, <*3>

$$d^* = \arg\max \{ EC(d_{\emptyset\emptyset}) , EC(d_{1\emptyset}) , EC(d_{kk}) \} \tag{6}$$

where

$$1 = \arg\max_j \{ C_j (P_{j\emptyset\emptyset} - P_{jj\emptyset}) + \bar{C}_j (\bar{P}_{j\emptyset\emptyset} - \bar{P}_{jj\emptyset}) - T_{j\emptyset}\} \tag{7}$$

$$k = \arg\max_j \{C_j(P_{j\emptyset\emptyset}-P_{jjj})+\bar{C}_j(\bar{P}_{j\emptyset\emptyset}-\bar{P}_{jjj})+(PP)_j P_{jjj}(C_j-\bar{C}_j)-T_{jj} \tag{8}$$

<*3>. The notation arg. min. implies $d^*=d_{\emptyset\emptyset}$ or $d_{1\emptyset}$ or d_{kk} depending on which of the three values $EC(d_{\emptyset\emptyset})$, $EC(d_{1\emptyset})$, $EC(d_{kk})$ is the smallest. Here $d_{1\emptyset}$ is the best monitoring decision and d_{kk} is the best patching decision.

Consider a specialization of the above decision problem where the probabilities P_{ijk} are assumed to be independent of the decisions d_{jk} (that is, $P_{ijk} = P_i$), the costs $\bar{C}_i$ and T_{ij} are all zero, and the patch success probabilities $(PP)_i=1$ for each subsystem RPV. Then the optimal decision is

$$d^* = d_{jj}$$

where

$$j = \arg\max_i \; [P_i \; C_i]$$

This is the result obtained by Carbonell(1971).

An implicit assumption made in the computation of expected cost in the combined monitoring and patching decision problem is that the costs are constant over the entire sets $\bar{e}^i_T$ and e^i_T. This assumption is easily dropped when non-constant cost functions are desired, e.g.,

$$C(e^i) = e^{i'} \; M \; e^i$$

In such a case, $P_{ijk}C_i$ in the above analysis will be replaced by

$$\int_{\bar{e}^i_T} C(e^i) \;\; N(\hat{e}^i, \Sigma^{ii}) \; d\, e^i$$

This would yield $P_{ijk}C_i$ as a function of $\hat{e}^i$ and Σ^{ii} and appears amenable for computations. Non-constant cost functions of the quadratic variety will be investigated further, if necessary.

We close this appendix, with an example of a piecewise-constant cost function that appears meaningful for the N-RPV system under study. Recall from appendix A that the first two components of e^i are:

e^i_1 = ground speed error (along track)

e^i_2 = cross-track error

One choice for the piecewise-constant cost function is:

$$
\begin{aligned}
C(e^i) &= 1 \quad \text{if} \quad |e_2^i| > e_{2T}^i = 250 \\
&= 0 \quad \text{if} \quad |e_2^i| \leq 250
\end{aligned}
$$

APPENDIX C: CHANGES REQUIRED IN SAINT MODEL

Tasks 1-4

After initialization, all operators proceed to new task 13.

Task 5 (MODRF 13, USERF 1)

No changes

Task 8 (USERF 3)

Now incorporated into new task 13.

Task 10 (USERF 4)

Now incorporated into new task 13.

Task 13 (USERF 7)

Replaced by new task 13. This new task is released from tasks 1,2,3,4,21,31,47,48,54,55,57,58,59,64,67,68,77,78, and 84, as well as from new task 13 itself. It incorporates the operator decisions in tasks 91,8,10,13,16, and 18, and combines all of these into a central decision loop. Much of the code in USERF 2,3,4,5,7 and 8 and in MODRF 1 and 5 can be utilized in the development of this new task.

Task 16 (MODRF 1, USERF8)

Now incorporated into new task 13.

Task 18 (MODRF 5)

Now incorpoated into new task 13.

Task 20 (USERF 22)

Must be changed so that single fuel check index (FLAST) now becomes an array, FLAST (I), with an entry for each separate RPV.

Task 21 (USERF 10)

Must be changed so that a fuel check is made only for the particular RPV specified as the task is called. The task performance time should therefore be sampled from a normal distribution with mean 3 and standard deviation 1.

As before, the operator proceeds to task 56 if fuel conservation is required. If not, the operator is directed back to new task 13.

Task 23

No changes

Task 24 (MODRF 12)

No changes

Task 27 (MODRF 8, MODRF 11)

No changes

Task 29 (MODRF 9, USERF 23)

No changes

Task 31 (MODRF 10, USERF 12)

MODRF 10 should be changed to base its decisions on RPV type rather than RPV number. In both MODRF 10 and USERF 12, the tests based on the ISTER array should be eliminated and replaced with objective tests. We believe that the test for missed handoffs outlined in the flowchart for new Task 13 will adequately cover the problem of determining when a handoff has been missed. The branching from this task should be based on the type of RPV handed off rather than on operator number. If the RPV was type S, the operator should proceed to new task 13. If the RPV type was E or L, he should proceed to task 34 to await handback.

Task 40

No changes

Task 41 (MODRF 17)

No changes

Task 42

No changes

Task 43 (USERF 24)

An objective test for missed popdowns must be developed to eliminate the use of the ISTER array in USERF 24.

Task 47 (MODRF 16)

All operators should proceed from this task to new task 13.

Task 48 (MODRF 17, USERF 26)

All operators should proceed from this task to new task 13.

Task 52

No changes

Task 53 (USERF 27, USERF 15)

No changes

Task 54 (MODRF 15)

All operators should proceed from this task to new task 13.

Task 55 (MODRF 15)

Task time should be computed according to the revised procedure presented in Section 3.3. All operators should proceed from this task to new task 13.

Task 56 (MODRF 18)

No changes

Task 57

The operator should proceed from this task to new task 13.

Task 58 (MODRF 19, USERF 16)

If malfunction has already been corrected, operator should return to new task 13.

Task 59 (USERF 30)

Operator should proceed after this task to new task 13.

Task 62

No changes

Task 63

No changes

Task 64 (USERF 17)

If no reroute is to be attempted, the operator should proceed to new task 13.

Task 65 (MODRF 21)

No changes

Task 66

No changes

Task 67

All operators should proceed from this task to new task 13.

Task 68 (USERF 18)

USERF 18 should be expanded to include the seach for an appropriate RPV for rerouting. This search, originally imbedded within task 8, has been eliminated from new task 13.

Task 69

No changes

Task 70

No changes

Task 71 (MODRF 1)

No changes

Task 73

No changes

Task 74 (MODRF 2, USERF 19)

No changes

Task 76 (USERF 20)

No changes

Task 77 (MODRF 22)

The operator should proceed from this task to new task 13.

Task 78 (MODRF 23)

The operator should proceed from this task to new task 13.

Task 79 (USERF 21)

No changes

Task 80 (MODRF 24)

The operator should proceed from this task to new task 13.

Task 81 (MODRF 1)

No changes

Task 83

No changes

Task 84 (MODRF 2, USERF 13)

If reprogram is unsuccessful, operator should proceed to new task 13.

Task 86 (MODRF 3, USERF 31)

No changes

Task 87 (MODRF 25)

No changes

Task 88

No changes

Task 89 (MODRF 20)

No changes

Task 90

No changes

Task 91 (USERF 2)

Now incorporated with new task 13.

Task 92 (MODRF 26)

No changes

Task 93 (MODRF 27)

No changes

Task 94

No changes

Task 95

No changes

Task 96 (USERF 31)

No changes

Task 97 (MODRF 29)

No changes

APPENDIX D: PROPOSED PARAMETER VALUES FOR BOTTOM-UP MODEL
(reference: Figures 2, 3, 4, and 5)

Process/Code	Variable = value
Action limits	
e_2	IAL = 6
b_4	IAL = 3 DAL = 1
b_6	IAL = 3 DAL = 1
Pop-up threshold	
b_0	t_1 = 30 sec (6 frames)
Final adjustment threshold	
b_4	N = 15 sec (3 frames)
LATDEV patching (see Figure 5)	A_i: Distance to display = 8 inches Distance to button console = 3 inches
	$\left.\begin{array}{l} W_i \\ a_i \\ b_i \end{array}\right]$ defined by user

APPENDIX E: EXECUTIVE SUMMARY OF REPORT 3446

This report describes the results and recommendations derived from an extensive survey of existing human performance models and modelling approaches applicable to the design and evaluation of large-scale command and control systems. The focus is on models derived from a control- and decision-theoretic framework, the modelling literature in human information processing, and the collection of models and data-bank formulations originally derived from the reliability and network-simulation literature.

The most successful modelling efforts seem to have grown out of situations where formal models of the task environment are well developed, such as in feedback control tasks, detection tasks, and well-defined probabilistic decision-making tasks. Further, in these areas, the most successful of these models arise when the researcher can express formal criteria of optimal performance as reflected in the optimal control formulation of manual control or the ideal observer in target detection and recognition tasks. These observations emphasize the importance to successful modelling efforts of being able to express the goals and success criteria used by human operators in formal quantitative terms. One difficulty for modelling behavior in more complex procedural tasks arises from the inherently multi-dimensional, multi-level, time-varying array of criteria and strategies that an operator applies in accomplishing these tasks.

It is interesting to note that the optimal control model and those information-processing models derived from a decision-theoretic framework are mutually compatible; this suggests the possibility of integrating and generalizing them to provide a single modelling framework that could be applied to vehicle control, supervisory monitoring, surveillance, signal identification, and decision making, all of which are tasks of major interest in military system design and evaluation.

In the area of intellectual performance, however, modelling efforts have not produced practically useful results, either in areas where an explicit algorithm might be specified or with respect to general problem-solving performance, where a wide range of performance strategies are available. To represent these kinds of performance will require either structuring the problem so that results are not sensitive to differences in strategy or resorting to atheoretic representations derived from empirical measurements obtained in the specific task context.

In addition to the substantive reviews of modelling approaches, several methodological issues have been identified:

(1) The problem of validation of models of the scope considered here is a difficult one, requiring further research.

(2) Models exist at many different levels of specificity. A substantive issue concerns the identification of the level of specificity at which to define a model in relation to the goals of system performance prediction desired. Depending on the level of specificity that is appropriate, one must consider whether to take a top-down or a bottom-up approach. A top-down approach begins with goals of performance prediction and represents performance only down to level required to meet these goals. A bottom-up approach begins by defining the elemental components of human performance and synthesizes them into a model that predicts the desired aspects of performance.

(3) Bottom-up approaches to modelling continually face the issue of how to combine sub-task or task models into higher-level structures in such a way that the potential interactions resulting from their combination are accounted for. Additive combinations of component response times and multiplicative combinations of respo[illegible] accuracies are frequently not valid, and their applicability must be evaluated in each new synthesis.

(4) The current state of theory and understanding of human Performance is inadequate to represent many kinds of behavior observed in real task situations. The model developer is left with many arbitrary parameters that must be defined on the basis of observed performance. When the number of such free parameters approaches the number of performance measures to be predicted, the predictive power of the model is severely compromised.

On the basis of our review, recommendations are introduced for further research and development of large scale systems modelling efforts. These recommendations include:

(1) <u>Development of a test-bed facility</u> in which to evaluate alternative model formulations of common task environments and to conduct empirical validation studies to compare model predictions with actual human performance.

(2) <u>Methodological research</u> on (a) the implications of combining sub-task or information processing component models on system performance in the aggregate, (b) validation of large scale simulation models, and (c) development of guidelines for the acceptable number of free parameters in useful predictive models.

(3) Recommended further model development in topical areas of high priority for representation of command and control systems. Two such areas are supervisory control and monitoring and the prediction of team performance on the basis of performance of individual team members.

(4) Advancing the state-of-the-art with respect to the specific modelling approaches discussed in the body of the report.

From an overall perspective, we believe that integrative models of human performance compatible with the requirements for representing command and control system performance do not exist at the present time. What is available is a collection of component models and modelling principles formulated in a variety of frameworks, which might be drawn together to build an eclectic model for particular task situations of interest. On the basis of our present level of understanding, assembly of the components will call for substantial effort and is likely to require many assumptions about particular aspects of performance. If one is to have confidence in the product generated in this way, several iterative validation steps will be required.